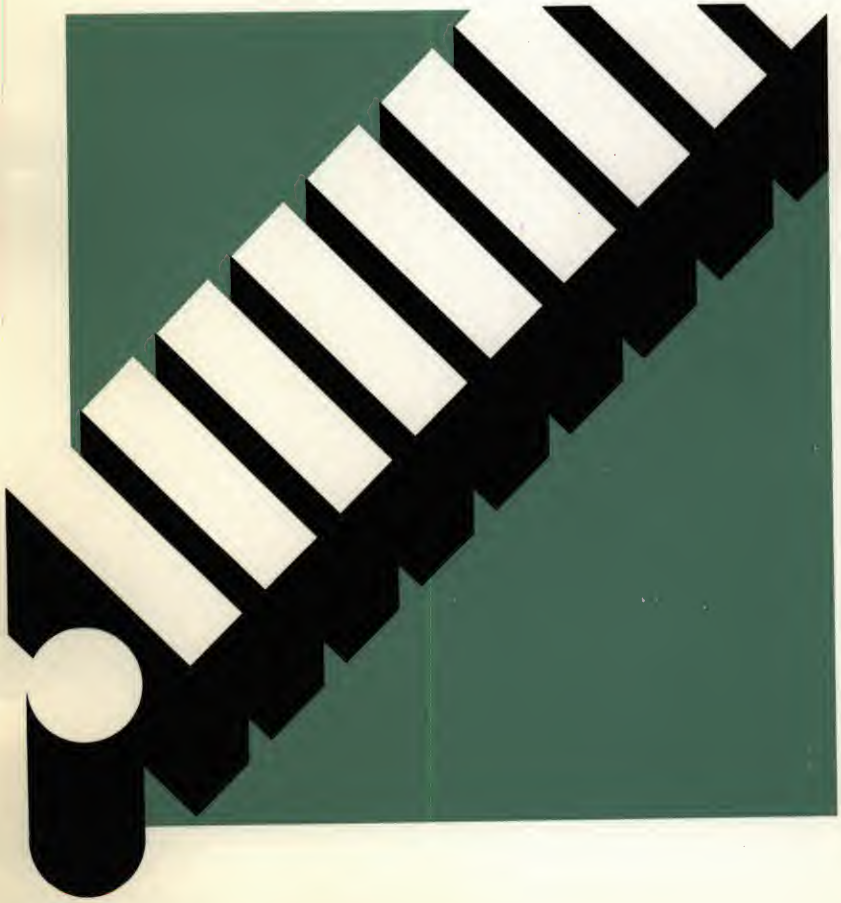
ASHTON·TATE

Naslaghandboek van dBASE IV



Naslagshandboek van dBASE IV

Naslagshandboek van dBASE IV

Inhoud

Basisbegrippen

Commando's

SET-commando's

Functies

Systeemgeheugenvariabelen

SQL-commando's

SQL-catalogi

Bijlagen

Index

Inleiding

Het *Naslaghandboek* is een encyclopedie van de commando's, de functies en de systeemgeheugenvariabelen van dBASE IV. Deze handleiding is bestemd voor degenen die *dBASE IV voor beginners* of *Menu's van dBASE IV* hebben doorgenomen. Ook als u al vertrouwd bent met het programmeren in dBASE of met database-beheer, is deze handleiding een nuttige naslaghandleiding.

Als u een tamelijk tot zeer gevorderde gebruiker van dBASE bent en u hebt de Developer's Edition gekocht, kunt u wellicht ook *Programmeren met dBASE IV* en het boekje *Voorbeeldprogramma's* raadplegen.

Deze handleiding bestaat uit zeven hoofdstukken, acht bijlagen en een index.

Hoofdstuk 1: Basisbegrippen

In dit hoofdstuk worden de basisbegrippen van dBASE IV besproken: het gebruik van commando's, SET-commando's, functies en systeemgeheugenvariabelen. Verder worden de componenten van de programmeertaal van dBASE IV behandeld en wordt uitgelegd hoe u met deze componenten een commandoregel samenstelt.

Hoofdstuk 2: Commando's

Dit hoofdstuk bevat een alfabetische lijst van de commando's die u in de programmeertaal van dBASE IV kunt gebruiken. Bij elk trefwoord is het basismodel van de syntaxis opgenomen en worden opmerkingen gegeven over het gebruik van het commando. Bij veel commando's worden ook tips, voorbeelden en verwijzingen naar andere commando's en functies gegeven.

De database-bestanden en de indexbestanden die in de voorbeelden worden gebruikt, worden in bijlage C, "Voorbeeldbestanden", opgesomd. Verder bevat bijlage C een volledige uitdraai van Menus.prg. Aan dit programma zijn in dit hoofdstuk verscheidene voorbeelden ontleend.

Hoofdstuk 3: SET-commando's

De SET-commando's vormen een deelverzameling van de dBASE IV-commando's en worden in dit hoofdstuk in alfabetische volgorde opgesomd. Met de SET-commando's kunt u bepaalde instellingen opgeven voor de manier waarop dBASE de gegevens op het scherm weergeeft of op de printer afdrukt. Met deze instellingen kunt u ook een omgeving maken die invloed heeft op de manier waarop andere commando's en functies werken.

Hoofdstuk 4: Functies

Hoofdstuk 4 biedt over functies dezelfde soort informatie die in hoofdstuk 2 en 3 over commando's wordt gegeven. Alle functies staan in alfabetische volgorde en worden uitgebreid beschreven. Deze beschrijving omvat onder andere een definitie, de syntaxis, het gebruik, voorbeelden en handige tips.

Hoofdstuk 5: Systeemgeheugenvariabelen

Systeemgeheugenvariabelen zijn speciale geheugenvariabelen die in dBASE IV zijn gereserveerd voor het vastleggen van gegevens over het afdrukken en de indeling van de afdruk. Dit hoofdstuk bevat informatie over het aanpassen van de variabelen voor de afdruktaken.

Hoofdstuk 6: SQL-commando's

In hoofdstuk 6 treft u een overzicht van de SQL-commando's aan. Ieder SQL-commando wordt besproken aan de hand van de syntaxis, een omschrijving en voorbeelden.

Hoofdstuk 7: SQL-catalogustabellen

In hoofdstuk 7 vindt u een omschrijving van de SQL-catalogustabellen.

Bijlagen

In de bijlagen staat het volgende:

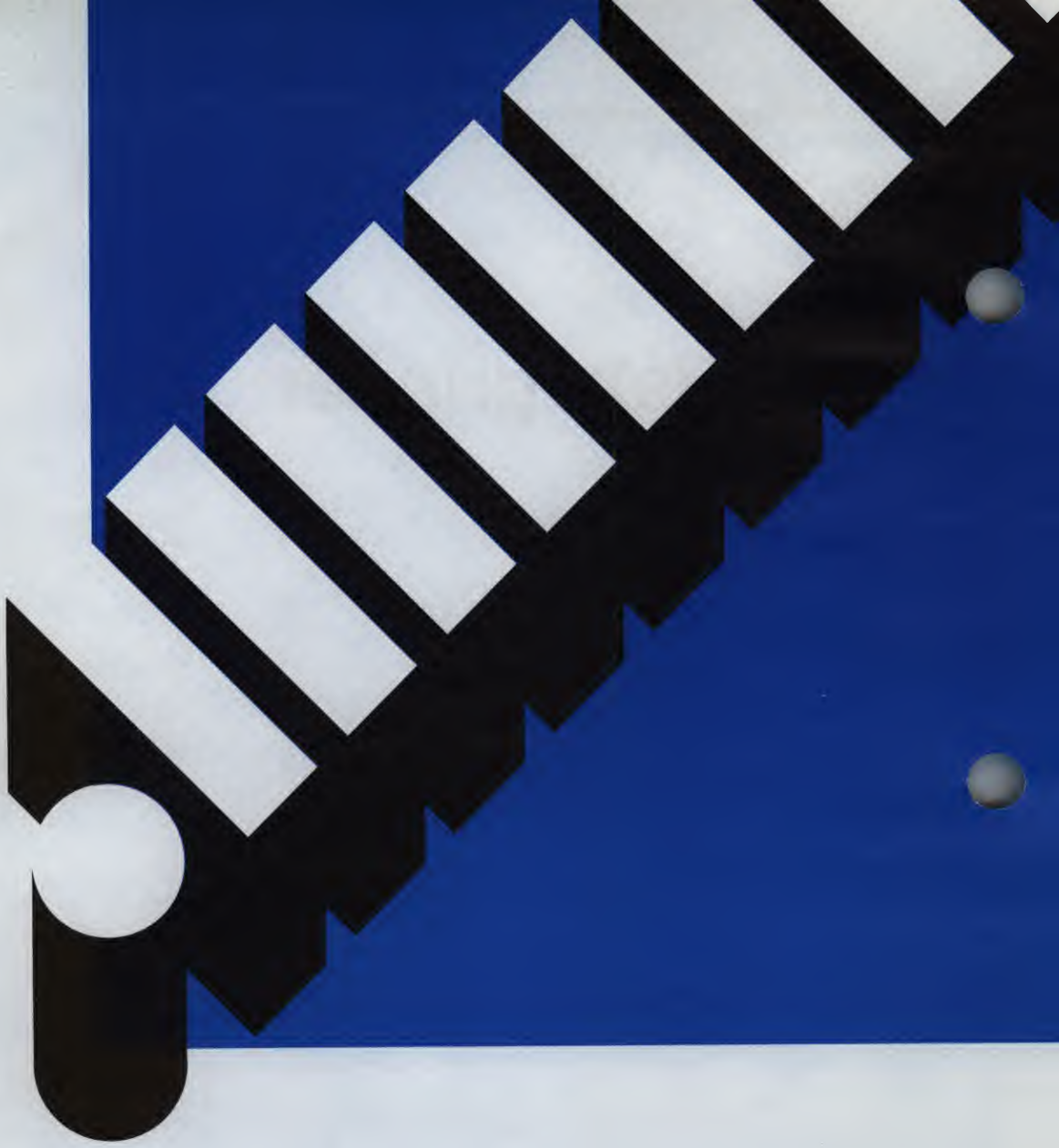
- een lijst met foutmeldingen en de betreffende codegetallen. Tevens worden de foutmeldingen uitgelegd en worden de mogelijke oorzaken genoemd.
- technische specificaties van dBASE IV
- de voorbeeldbestanden die voor alle voorbeelden in deze handleiding zijn gebruikt
- een samenvatting van de bestandstypen die dBASE IV gebruikt en maakt en de extensies die aan de bestandsnaam worden toegevoegd wanneer deze bestanden op schijf worden opgeslagen
- opmerkingen over de structuur van een database-bestand (.dbf)
- een lijst van de beschikbare printerstuurprogramma's en lettertypen die door deze programma's worden ondersteund
- de ASCII-tabel die de hexadecimale en de decimale equivalenten van elk teken bevat
- de commando's en functies van dBASE IV die u in de SQL-modus kunt gebruiken.

Index

In de index staan verwijzingen naar alle begrippen, commando's, functies en variabelen die in het *Naslaghandboek* zijn besproken.

Naslaghandboek van dBASE IV

Basisbegrippen



i 1 2 3 4 5 6 7 A B

C D E F G H In

Basisbegrippen

Via het Control Center kunt u gebruik maken van de commando's, SET-commando's, functies en systeemgeheugenvariabelen zonder de bijzonderheden van de werking daarvan te kennen. Als u echter vertrouwd bent met de componenten van de programmeertaal van dBASE IV, kunt u ook zelf *commandoregels* samenstellen vanaf de *commandostip* of in een *programma*.

In het kort

In dit hoofdstuk treft u de basisbegrippen aan die u moet kennen om een dBASE IV-commandoregel samen te stellen. Een commandoregel is een afgeronde opdracht waarmee de dBASE IV-processor handelingen uitvoert op gegevensonderdelen. Deze commandoregel kan een *commando* (of een *SET-commando*), *functies* en *systeemgeheugenvariabelen* bevatten. Over deze onderwerpen vindt u meer informatie in respectievelijk hoofdstuk 2, 3, 4 en 5.

Een commandoregel kunt u vanaf de *commandostip* of als onderdeel van een *programma* invoeren.

Componenten van de programmeertaal van dBASE IV

In de commandoregel speelt elke component van de programmeertaal een rol:

- Commando's zijn werkwoorden die de dBASE-processor opdracht geven een bepaalde handeling uit te voeren. Hoewel een commandoregel desgewenst uit meerdere taalcomponenten kan bestaan, mag deze slechts één commando bevatten. In sommige gevallen is het werkwoord dat voor het commando wordt gebruikt, impliciet aanwezig in de commandoregel, zoals bij het commando STORE. De commandoregels:

```
. STORE 5 TO x
```

en

```
. x = 5
```

zijn gelijk, maar in de tweede regel is het commando STORE impliciet aanwezig. Aangezien elke commandoregel een werkwoord als commando moet bevatten, wordt de term *commandoregel* vaak aangeduid met *commando*.

- SET-commando's vormen een deelverzameling commando's waarmee meestal de omgeving wordt ingesteld waarin de processor actief is. Met bijvoorbeeld SET COLOR worden de kleuren van het beeldscherm ingesteld en worden andere commando's met schermuitvoer beïnvloed.

- Functies kunnen op verschillende manieren worden gebruikt:

  1. Sommige functies, zoals PI(), leveren altijd een constante waarde op.

  2. Sommige functies zijn bijvoeglijke naamwoorden of bijwoorden waarmee een gegevensonderdeel wordt gewijzigd. Met bijvoorbeeld LOWER() worden hoofdletters in kleine letters omgezet.

  3. Andere functies bevatten een waarde of voorwaarde van dBASE IV die u kunt opvragen. De functie EOF() wordt waar (.T.) wanneer het einde van een database-bestand is bereikt en het commando

     ```
     . ? EOF()
     ```

     resulteert in

     ```
     .T.
     ```

  4. Sommige functies zijn formules die worden berekend aan de hand van de ingevoerde parameters. SQRT(x) bijvoorbeeld resulteert in de vierkantswortel van de ingevoerde parameter *x*.

  5. Door andere functies, zoals LOOKUP(), SEEK(), RLOCK() en FLOCK(), worden handelingen uitgevoerd die waarden opleveren.

- Systeemgeheugenvariabelen zijn instellingen waarmee het uiterlijk van de uitvoer naar het scherm of naar de printer wordt bestuurd. In eerdere versies van dBASE (dBASE II®, dBASE III® en dBASE III PLUS®) vormden commando's, SET-commando's en functies reeds componenten van de programmeertaal, maar systeemgeheugenvariabelen zijn nieuw in dBASE IV.

  De overeenkomst tussen systeemgeheugenvariabelen en SET-commando's is dat u hiermee meestal systeemparameters bestuurt en geen handelingen op gegevensonderdelen uitvoert. De overeenkomst met functies is dat u de waarden die de systeemgeheugenvariabelen bevatten, kunt opvragen. De overeenkomst met geheugenvariabelen is dat ze public of private kunnen zijn.

In dit boek zijn de commando's in hoofdletters afgedrukt, bijvoorbeeld LIST. De SET-commando's staan ook in hoofdletters en beginnen altijd met het woord SET, bijvoorbeeld SET DEVICE. De functies zijn ook in hoofdletters afgedrukt maar worden door haakjes afgesloten, bijvoorbeeld FOUND(). De systeemgeheugenvariabelen staan in kleine letters en beginnen met een onderstrepingsteken, bijvoorbeeld _padvance. Door deze typografische verschillen kunt u de verschillende termen gemakkelijker van elkaar onderscheiden: CHANGE() bijvoorbeeld is een functie, maar CHANGE is een commando.

Commandostip-interface

De commandostip-interface is de interactieve modus van dBASE IV. In deze modus kunt u een commandoregel invoeren die onmiddellijk wordt uitgevoerd. Deze modus wordt aangeduid door een stip op het scherm en heet daarom *commandostip*. Vanaf de commandostip kunt u sneller en soepeler werken dan wanneer u alleen vanuit het Control Center werkt.

In dBASE IV geeft u een commando door dit vanaf de stip te typen en daarna op ↵ te drukken. Elke regel die u invoert, kan uit maximaal 254 tekens bestaan.

Als u een lange commandoregel vanaf de stip wilt typen, drukt u op **Ctrl-Home** zodat er een *bewerkscherm* wordt geopend. Tijdens het typen van de commando's in het bewerkscherm hebt u de beschikking over alle voorzieningen van de tekstverwerker van dBASE IV en kunt u de gehele instructie zien zonder het scherm van het begin naar het einde van de commandoregel te hoeven verschuiven. Het maximum aantal tekens van de totale commandoregel in een bewerkscherm is 1024.

Commando's en bepaalde sleutelwoorden kunt u na de eerste vier letters afkorten (dit geldt niet voor SQL-commando's). U kunt bijvoorbeeld REPORT FORM afkorten als REPO FORM en MODIFY COMMAND als MODI COMM, maar bij de SQL-commando's mag dit niet.

Een commandoregel kunt u in hoofdletters, in kleine letters of in een combinatie hiervan invoeren. Tussen de woorden van een commandoregel kunt u een willekeurig aantal spaties opnemen. Elke spatie telt echter mee als één teken.

Commando's opnieuw invoeren

De commandostip heeft een geheugenbuffer, die het *logboek* wordt genoemd, waarin de commando's tijdens het invoeren automatisch worden opgeslagen. Hierdoor kunt u teruggaan naar een eerder ingevoerd commando en dit bewerken of uitvoeren. Bij de installatie van dBASE IV is de standaardinstelling van de logboekbuffer 20 commando's. Met SET HISTORY kunt u het standaardaantal van 20 opgeslagen commando's wijzigen in een getal tussen 0 en 16.000. U kunt deze standaardinstelling ook opnieuw configureren door het bestand Config.db te wijzigen (raadpleeg *Aan de slag met dBASE IV*).

Als u de commando's die al in de logboekbuffer zijn opgeslagen, wilt weergeven, drukt u bij de commandostip op ↑. De commando's worden dan één voor één in omgekeerde volgorde weergegeven. Met ↓ verplaatst u de cursor in de lijst van boven naar beneden. U kunt het opgeslagen commando bewerken en het commando uitvoeren door op ↵ te drukken. Met behulp van DISPLAY HISTORY en LIST HISTORY kunt u meer dan één commando tegelijkertijd bekijken.

SQL-interface

In de interactieve SQL-modus kunt u SQL-commandoregels op vrijwel dezelfde manier invoeren als waarop u een dBASE-commando invoert. Bij de commandostip kunt u de SQL-modus activeren met het commando SET SQL ON. Daarna verschijnt de SQL-aanduiding, SQL. U kunt de dBASE-modus activeren en de commandostip weer op het scherm weergeven door vanaf de SQL-aanduiding SET SQL OFF in te voeren.

Een SQL-commando kunt u rechtstreeks invoeren vanaf de SQL-aanduiding of in een bewerkvenster, net als in de dBASE-modus. U kunt ook de logboekbuffer gebruiken om SQL-commando's opnieuw invoeren. Voor het invoeren van SQL-commando's gelden dezelfde voorwaarden als voor dBASE-commando's. De enige uitzondering is dat de sleutelwoorden van de SQL-commando's niet mogen worden afgekort.

In de SQL-modus kunt u beschikken over de meeste commando's en functies van dBASE zodat u hiermee de werking van de SQL-commando's kunt aanvullen. Deze commando's en functies treft u aan in bijlage H.

Als u bijvoorbeeld rapporten en etiketten wilt afdrukken, kunt u de SQL-uitkomsten bewerken en opmaken met de dBASE-commando's. Met deze commando's kunt u ook bijvoorbeeld de SQL-omgeving definiëren. Verder kunt u met de dBASE-functies de manier beïnvloeden waarop de SQL-commando's worden verwerkt en hoe de uitkomsten worden weergegeven.

Voordat u SQL-commando's invoert voor tabellen, visies en andere SQL-objecten, moet u met het SQL-commando START DATABASE een database starten. In hoofdstuk 6 treft u informatie aan over het gebruik van SQL-commando's.

Bestandsnamen en aliassen

In bijlage D staan lijsten met de typen bestanden die dBASE IV maakt en gebruikt. Met behulp van aliassen kunt u de gegevens uit een aantal verschillende database-bestanden oproepen en relateren.

Bestandsnamen

Een bestandsnaam kan de werkelijke naam van een bestand zijn zoals dit is opgeslagen, maar ook een *indirecte verwijzing* naar de bestandsnaam. Verderop in dit gedeelte worden de indirecte verwijzingen naar bestanden behandeld. Wanneer u de naam van een bestand invoert, kunt u ook met de stationsspecificatie en het volledige pad aanduiden waar het bestand zich bevindt.

In dBASE IV kunt u alle geldige DOS-bestandsnamen gebruiken. Wanneer u een bestand opslaat, wordt door dBASE IV een extensie aan het bestand toegewezen waarmee wordt aangeduid welk type gegevens het bestand bevat. Door bijvoorbeeld het commando CREATE wordt een bestand met de extensie .dbf weggeschreven. Deze extensie is voor andere dBASE-commando's een aanduiding dat het bestand records met gegevens bevat. In bijlage D staat een volledige lijst van de extensies die door dBASE IV aan bestanden worden toegewezen.

OPMERKING
De bestandsnamen mogen geen DOS-namen voor randapparaten of poorten zijn. Raadpleeg het DOS-boek voor deze DOS-namen.

Indirecte verwijzingen naar bestanden vormen een belangrijke nieuwe voorziening in dBASE IV. Een indirecte verwijzing is een tekenuitdrukking die resulteert in een bestandsnaam en die u overal kunt gebruiken waar u een bestandsnaam moet opgeven. In deze uitdrukking moet u een operator gebruiken (meestal haakjes), zodat dBASE weet dat de tekenreeks een uitdrukking en geen letterlijke bestandsnaam is. U kunt bijvoorbeeld bij het commando CREATE een indirecte verwijzing in plaats van een bestandsnaam gebruiken. Als de variabele Mbestand de tekenreeks "Term" bevat, wordt met de commando's CREATE (Mbestand) en CREATE RTRIM(Mbestand) een database-bestand gemaakt met de naam Term. Deze commando's zijn gelijk aan CREATE Term. Met CREATE Mbestand+"01" wordt een database-bestand met de naam Term01 gemaakt.

Een indirecte verwijzing gebruikt u op dezelfde manier als het macrosubstitutie-teken, zoals CREATE &Mbestand, maar de indirecte verwijzing wordt veel sneller uitgevoerd.

In het volgende voorbeeld roept u met het eerste USE-commando tijdens de uitvoering de compiler aan terwijl u de compiler met het tweede USE-commando niet aanroept:

```
. Mbestand = "Afnemers"
. USE &Mbestand
. USE (Mbestand)
```

Bepaalde tekens die u in een bestandsnaam opneemt (bijvoorbeeld () en &) worden door dBASE IV geïnterpreteerd. Gebruik voor dergelijke tekens een indirecte verwijzing:

```
. CREATE "TEST()"
```

Met de commando's CREATE/MODIFY QUERY/VIEW, CREATE/MODIFY LABEL, CREATE/MODIFY REPORT, CREATE VIEW FROM ENVIRONMENT en SET CATALOG worden de namen van één of meer actieve bestanden opgeslagen, zodat deze bestanden op een later tijdstip automatisch kunnen worden geopend.

Met CREATE/MODIFY QUERY/VIEW wordt een query-bestand (.qbe) gemaakt waarmee u records, die aan de opgegeven voorwaarden voldoen, kunt ophalen. U kunt met dit commando ook een mutatie-query-bestand (.upd) maken waarmee de records van een database-bestand kunnen worden bijgewerkt. Wanneer u met CREATE/MODIFY REPORT of CREATE/MODIFY LABEL rapporten of etiketten samenstelt, kunt u de instellingen van systeemgeheugenvariabelen in een afdrukbestand voor formulieren (.prf) opslaan. De naam van het .prf-bestand wordt in het bijbehorende rapportbestand (.frm) of etiketbestand (.lbl) opgeslagen. Met CREATE VIEW FROM ENVIRONMENT wordt een visiebestand (.vue) samengesteld dat de namen van database-bestanden, indexbestanden en indelingsbestanden kan bevatten. Met SET CATALOG wordt een catalogusbestand (.cat) gemaakt dat de namen van database-bestanden en visiebestanden en de daarmee verbonden indexbestanden, indelingsbestanden, etiketbestanden en rapportbestanden bevat.

Soms worden de bestandsnamen die in de .qbe-, .upd-, .frm-, .lbl-, .vue- of .cat-bestanden zijn opgeslagen, met de bijbehorende stationsnaam en padnaam geschreven en soms is dit niet het geval. Het hangt er namelijk van af of de bestandsnaam in het standaardstation en in de actieve bestandsindex aanwezig is en of u tijdens de samenstelling van de .qbe-, .upd-, .frm-, .lbl-, .vue- of .cat-bestanden het station en het pad expliciet als onderdeel van de bestandsnaam hebt opgegeven.

Er zijn twee manieren om een station en een pad expliciet op te geven:

1. U geeft het station en het pad op als onderdeel van de commandoregel. Voorbeeld:

```
. USE C:\DBASE\MIJNDATA\Mijnbest
```

   Hiermee geeft u zowel het station, C:, als het pad, \DBASE\MIJNDATA, op waar het bestand zich bevindt.

2. Met de vraagclausule, een vraagteken, verplaatst u de cursor naar de positie waar het bestand is opgeslagen en kiest u het bestand. Het openen van een bestand op deze manier is vergelijkbaar met het openen van een bestand waarbij u het station en het pad expliciet als onderdeel van de bestandsnaam opgeeft.

dBASE IV bepaalt aan de hand van de volgende voorwaarden of de bestandsnaam met de volledige stationsnaam en padnaam moet worden opgeslagen:

1. Als u een bestand in het standaardstation en de actieve bestandsindex opent, worden de bestandsnamen met de eerder vermelde commando's zonder de gegevens over het station en de bestandsindex opgeslagen. Dit geldt ook wanneer u als onderdeel van de bestandsnaam een station en een pad opgeeft. Stel dat het standaardstation C: en de actieve bestandsindex \DBASE\MIJNDATA is. Als u met SET CATALOG een catalogus hebt geopend en met USE het bestand C:\DBASE\MIJNDATA\Mijnbest in gebruik hebt genomen, wordt het bestand als Mijnbest.dbf in de catalogus opgeslagen. De bestandsnaam wordt vanzelfsprekend ook als Mijnbest.dbf opgeslagen als u USE Mijnbest typt.

2. Een bestand dat zich niet in het standaardstation en de actieve bestandsindex bevindt, kunt u openen door het station en de bestandsindex als onderdeel van de bestandsnaam op te geven of door met SET PATH een zoekpad in te stellen.

   a. Wanneer u het station en het pad als onderdeel van de bestandsnaam opgeeft, worden de gegevens over het station en het pad als onderdeel van de bestandsnaam opgeslagen. Als het actieve station dus nog C: en de actieve bestandsindex nog \DBASE\MIJNDATA zijn en u USE A:\UWDATA\Uwbest typt, wordt het bestand als A:\UWDATA\Uwbest.dbf in de catalogus opgeslagen.

   b. Als u met SET PATH een zoekpad instelt, wordt de bestandsnaam opgeslagen zoals u deze invoert. Met de commando's

```
. SET PATH TO A:\UWDATA
. USE Uwbest
```

   wordt de bestandsnaam als Uwbest.dbf in de catalogus opgeslagen.

   Met de commando's

```
. SET PATH TO A:\UWDATA
. USE A:\UWDATA\Uwbest
```

   wordt de bestandsnaam als A:\UWDATA\Uwbest.dbf in de catalogus opgeslagen.

De .qbe-, .upd-, .frm-, .lbl-, .vue- en .cat-bestanden kunnen dus nooit bestandsnamen met stationsnamen en padnamen bevatten als het bestand zich in het standaardstation en de actieve bestandsindex bevindt. De .qbe-, .upd-, .frm-, .lbl-, .vue- en .cat-bestanden bevatten alleen bestandsnamen met stationsnamen en padnamen als het bestand zich in een ander station of in een andere bestandsindex bevindt *en* als u de stationsnaam en de padnaam expliciet opgeeft.

Aliassen

In dBASE IV kunt u maximaal tien database-bestanden tegelijk openen door aan elk geopend bestand een uniek werkgebied toe te wijzen.

De geopende database-bestanden worden intern door dBASE IV aangeduid met *aliassen*. Een alias kan een aliasnaam (die gelijk aan de bestandsnaam kan zijn), een werkgebiedletter of een werkgebiednummer zijn.

Als u met de ALIAS-optie van het commando USE een aliasnaam toewijst, kunt u in plaats van de bestandsnaam deze aliasnaam als afkorting gebruiken wanneer u verwijst naar het werkgebied. Als u met de ALIAS-optie van het commando USE geen aliasnaam toewijst, wordt door dBASE IV als standaard-aliasnaam de bestandsnaam gebruikt.

Als werkgebiedletters worden de letters A tot en met J of a tot en met j gebruikt. Werkgebied A is het eerste werkgebied en J is het tiende werkgebied.

Als werkgebiednummers worden de cijfers 1 tot en met 10 gebruikt. In eerdere versies van dBASE kon u alleen bij het commando SELECT werkgebiednummers gebruiken. In dBASE IV kunt u bij alle commando's die met aliassen werken, een werkgebiednummer gebruiken.

Een aantal geldige bestandsnamen, zoals X-y, kan niet als alias worden gebruikt omdat ze tekens bevatten die in dBASE voor andere doeleinden zijn gereserveerd. Als een bestandsnaam tekens bevat waardoor deze niet als standaard-aliasnaam kan worden gebruikt en u met de ALIAS-optie van het commando USE geen andere aliasnaam toewijst, wordt door dBASE IV een letteralias (A tot en met J) als standaard-aliasnaam toegewezen.

OPMERKING

U kunt een indirecte verwijzing naar een alias gebruiken, behalve wanneer de alias als prefix voor een veldnaam wordt gebruikt. Als u in de tekenreeks van de alias een operator opneemt (meestal haakjes), weet dBASE IV dat de tekenreeks een uitdrukking en niet de letterlijke alias is. Voorbeeld:

```
. Expwa = 3
. GO 5 IN (Expwa)
```

Hiermee wordt de recordwijzer op het vijfde record van werkgebied 3 geplaatst.

```
. ? RECCOUNT("Klant")
```

Dit commando levert het aantal records in het bestand Klant op, dat in een ander werkgebied is geopend.

Het gebruik van indirecte verwijzingen naar aliassen is vergelijkbaar met het macrosubstitutie-teken, maar de werking van indirecte verwijzingen is veel sneller.

Programma's en procedures

Als u dezelfde serie commando's herhaaldelijk uitvoert en u dit proces wilt automatiseren, kunt u deze opdrachten in een programmabestand opslaan.

Programma's

Een programma is een serie dBASE IV-commando's die zijn opgeslagen in een schijfbestand. Wanneer u het programmabestand uitvoert, worden de commando's in het programma op dezelfde manier uitgevoerd als wanneer u deze vanaf de commandostip typt.

U kunt voor het schrijven en opslaan van programma's gebruik maken van de tekstverwerker van dBASE IV die u oproept met MODIFY COMMAND. Met MODIFY COMMAND wordt een schijfbestand met de extensie .prg en voor een SQL-programma met de extensie .prs gemaakt.

Wanneer u een programmabestand maakt, drukt u op ↵ als u het einde van een commandoregel wilt aanduiden. Wanneer u een commando over meerdere regels wilt verdelen, typt u aan het einde van elke regel een puntkomma behalve aan het einde van de laatste regel.

Het programmabestand kunt u met het commando DO uitvoeren. Voordat DO het programmabestand uitvoert, worden de commando's naar *objectcode* gecompileerd. De objectcode wordt veel sneller uitgevoerd dan de oorspronkelijke *broncode* in het .prg- of .prs-bestand. Door DO wordt de objectcode naar een schijfbestand met de extensie .dbo weggeschreven. U kunt ook met het commando COMPILE een objectbestand genereren zonder het programma uit te voeren. Tijdens de compilatie wordt u door dBASE IV attent gemaakt op eventuele fouten die worden aangetroffen.

Procedures

Een programma kan zijn samengesteld uit één of meer routines. Deze worden *procedures* genoemd. Elke procedure voert meestal één basistaak uit en kan vanuit andere procedures of met een DO-commando vanaf de commandostip worden opgeroepen. Nadat een procedure de taak heeft uitgevoerd, wordt de besturing weer overgedragen aan het programma, aan de procedure van waaruit deze procedure werd aangeroepen of aan de commandostip of het Control Center.

In eerdere versies van dBASE bevonden de procedures zich meestal in een afzonderlijk bestand dat een *procedurebestand* werd genoemd. U kon met het commando SET PROCEDURE per keer één procedurebestand met maximaal 32 procedures openen.

dBASE IV verwerkt procedures op een andere manier. U kunt de procedures direct in een programmabestand opnemen of in een afzonderlijk procedurebestand plaatsen. Het programmabestand of het procedurebestand kan zoveel procedures bevatten als het RAM-geheugen toelaat (met maximaal 963 procedures per bestand). Elke procedure moet beginnen met het sleutelwoord PROCEDURE.

dBASE IV houdt aan het begin van elk objectbestand (.dbo) een procedurelijst bij. dBASE IV beschouwt het hoofdprogramma zelf ook als een procedure door daaraan een standaard-procedurenaam te geven die overeenkomt met de bestandsnaam van het bronprogramma. Deze procedurenaam is de eerste naam van de procedurelijst. Alle volgende procedures die zich in het actieve bronbestand of in een afzonderlijk procedurebestand bevinden, worden aan de lijst toegevoegd wanneer het bronbestand naar een objectbestand wordt gecompileerd.

Stel dat u het volgende programma met de naam Centraal hebt:

```
*Centraal.prg
<commando's>
DO A
DO B
DO C
RETURN


PROCEDURE A
<commando's>
RETURN


PROCEDURE B
<commando's>
RETURN


PROCEDURE C
<commando's>
RETURN
```

dBASE IV neemt in de procedurelijst vier procedures voor het gecompileerde objectbestand op: Centraal (de standaard-procedurenaam), A, B en C. Zoals u ziet, wordt de standaard-procedurenaam alleen aan de code aan het begin van een programmabestand toegewezen. Een eventuele losse code na RETURN en voor PROCEDURE wordt wel gecompileerd maar veroorzaakt tijdens de compilatie waarschuwingsmeldingen. Aangezien deze code niet door het DO-commando wordt uitgevoerd, controleert de compiler of deze code in de programmabestanden en in de objectcodebestanden moet worden ingebed.

Door de procedures in het hoofdprogramma te plaatsen, hebt u niet langer allerlei afzonderlijke procedurebestanden nodig. De programma's worden hierdoor sneller uitgevoerd omdat dBASE IV de procedures niet meer hoeft te openen en te sluiten voordat deze worden uitgevoerd. U kunt voor procedures die niet met het DO-commando kunnen worden geactiveerd, nog steeds SET PROCEDURE TO <procedure bestandsnaam> gebruiken.

Wanneer dBASE IV het commando DO <procedurenaam> in een programmacode aantreft, wordt de benoemde procedure in deze volgorde gezocht:

1. Zoek het objectbestand (.dbo) dat op dit moment wordt uitgevoerd.

2. Zoek in het SET PROCEDURE-bestand, indien er een actief is, naar de eerste procedure met die naam.

3. Zoek de overige geopende objectbestanden (het laatst geopende het eerst).

4. Zoek en open een objectbestand (.dbo) met die naam.

5. Zoek en compileer een programmabestand (.prg) met die naam.

6. Zoek en compileer een SQL-programma (.prs) met die naam.

Aangezien dBASE IV in deze volgorde zoekt, kunt u procedures met dezelfde naam voor elkaar verbergen. U kunt ook nog steeds met het commando SET PROCEDURE een procedurebestand openen wanneer u met het commando DO een zoekopdracht uitvoert.

De beperkingen van dBASE IV voor procedures zijn:

- per procedure 64 Kb aan gecompileerde code

- 32 actieve objectbestanden (.dbo), inclusief een bestand dat met SET PROCEDURE is geopend

- 963 procedures per objectbestand

Sommige dBASE IV-commando's compileren ook objectcode uit broncode die door een ontwerpscherm is gegenereerd. Met bijvoorbeeld REPORT FORM kan een .frg-bestand dat door CREATE REPORT is gemaakt, naar een .fro-bestand met objectcode worden gecompileerd. Aangezien dBASE IV procedures in de vorm van een objectbestand verwerkt, wordt er geen onderscheid gemaakt tussen de procedures die met DO, COMPILE en DBLINK zijn gemaakt. Tevens wordt er geen onderscheid gemaakt tussen de objectcode in indelingsbestanden (.fmo), rapportbestanden (.fro), etiketbestanden (.lbo) of query-bestanden (.qbo). Bovendien bestaat er geen verschil tussen objectcode die uit .prg-bestanden wordt gegenereerd en die uit SQL-programmabestanden (.prs) wordt gegenereerd. Alle objectcodeprocedures kunnen worden opgeroepen en aan de procedurelijst worden toegevoegd.

Commando's gebruiken

In dit gedeelte worden de verplichte en de optionele onderdelen van een commandoregel behandeld en worden regels gegeven voor het samenstellen van uitdrukkingen.

De structuur van een commandoregel wordt de *syntaxis* genoemd. Elke commandoregel begint met een werkwoord. Veel commando's bevatten bovendien een of meer clausules waarmee het commando aan specifieke behoeften kan worden aangepast. De algemene syntaxis van een commando wordt verderop beschreven.

OPMERKING
Er zijn veel uitzonderingen op het algemene model van de syntaxis dat verderop wordt gegeven. De opties in het model worden niet voor alle commando's gebruikt. De uitzonderingen staan in de alfabetische lijsten. Lees voordat u een component van de programmeertaal gebruikt, alle omschrijvingen bij de trefwoorden in de hoofdstukken hierna zorgvuldig door.

<commando> [<uitdrukkingenlijst>] [<bereik>]
[FOR <voorwaarde>] [WHILE <voorwaarde>] [TO FILE <bestandsnaam>
/TO PRINTER/ TO ARRAY <reekslijst!/TO <geheugenvariabele>]
[ALL [LIKE/EXCEPT <structuur>]] [IN <alias>]

<commando> is de naam van het dBASE-commando.

Rechte haken ([ ]) duiden aan dat het onderdeel optioneel is.

Punthaken (> <) duiden aan dat u een bepaalde waarde van het type, dat voor het onderdeel tussen de punthaken is vereist, moet invoeren.

Een schuine streep (/) duidt een of/of-keuze aan.

OPMERKING
Wanneer u een commando invoert, typt u geen rechte haken, punthaken of schuine strepen.

<lijst> is een groep gelijksoortige onderdelen die door komma's van elkaar worden gescheiden.

[<uitdrukkingenlijst>] staat voor één of meer uitdrukkingen die door komma's worden gescheiden. De uitdrukkingen hoeven niet van hetzelfde gegevenstype te zijn (raadpleeg het gedeelte "Uitdrukkingen" hierna).

[<bereik>] duidt het aantal records aan dat het commando kan oproepen. De sleutelwoorden voor het bereik zijn:

- met RECORD <n> roept u één record met het betreffende getal op
- met NEXT <n> roept u *n* records op, te beginnen met het actieve record
- met ALL roept u alle records in de database op
- met REST roept u alle records vanaf het actieve record tot aan het einde van het bestand op

Als in een commando een FOR- of WHILE-clausule wordt gebruikt, fungeren de voorwaarden die u in deze clausules opgeeft echter als beperkingen binnen <bereik>.

<voorwaarde> is een vergelijking tussen twee of meer onderdelen, zoals *Naam = "Smit"* of een logische opdracht zoals .NOT. EOF().

Door [FOR <voorwaarde>] weet dBASE IV dat het commando alleen van toepassing is op records die aan de voorwaarde voldoen. Als u FOR gebruikt, verplaatst dBASE IV de recordwijzer terug naar het begin van het bestand en wordt elk record met de voorwaarde van FOR vergeleken.

Met [WHILE <voorwaarde>] start u de verwerking met het actieve record in het database-bestand. De verwerking wordt voortgezet met elk volgend record zolang de voorwaarde waar (.T.) is.

Met [TO ...] bestuurt u de uitvoer van het commando. Met bepaalde commando's kunt u uitvoer naar een bestand, een printer, een opgegeven reeks of een geheugenvariabele zenden.

Geheugenvariabelen (of *variabelen*) zijn gegevenswaarden die tijdelijk in het RAM-geheugen worden opgeslagen. Aan elke waarde wijst u een naam toe zodat u deze later aan de hand van de naam weer uit het geheugen kunt ophalen. Met deze waarden kunt u berekeningen, vergelijkingen en andere bewerkingen uitvoeren. U kunt met één van de volgende commando's geheugenvariabelen maken: ACCEPT, AVERAGE, CALCULATE, COUNT, INPUT, PARAMETERS, PRIVATE, PUBLIC, STORE, SUM en WAIT.

Door het commando DECLARE wordt een speciale set geheugenvariabelen gemaakt die een *reeks* wordt genoemd. Een reeks is een eendimensionale of tweedimensionale tabel van waarden die in het geheugen zijn opgeslagen. Elke invoer in de reeks wordt een *element* genoemd en elk element in een reeks kan als een geheugenvariabele worden beschouwd en in een uitdrukking worden gebruikt.

Met commando's waarmee u gegevens naar een geheugenvariabele kunt uitvoeren, kunt u ook gegevens naar een reekselement uitvoeren.

Door [ALL [LIKE/EXCEPT <structuur>]] weet dBASE IV welke bestanden, velden of geheugenvariabelen die overeenkomen met de *structuur*, moeten worden opgenomen of uitgesloten. De structuur is een algemeen patroon waarmee bestandsnamen, velden of geheugenvariabelen kunnen overeenkomen. In de structuur kunt u het vraagteken (?) en de asterisk (*) als jokers gebruiken. Een ? staat voor één willekeurig teken en een * voor een groep willekeurige tekens.

Met [IN <alias>] kunt u het database-bestand in een ander werkgebied verwerken zonder dit met SELECT als actief werkgebied te kiezen. De IN-clausule kan de aliasnaam, de aliasletter, het aliasgetal of een uitdrukking die een aliasnaam, aliasletter of aliasgetal berekent, bevatten. Voor het USE-commando is echter in de IN-clausule een werkgebiednummer vereist omdat er pas na het openen van het bestand een andere alias aanwezig is.

<bestandsnaam> kan de werkelijke naam van een bestand zijn waaronder dit is opgeslagen of een indirecte verwijzing naar de bestandsnaam. Bestandsnamen en het gebruik van indirecte verwijzingen zijn in dit hoofdstuk al eerder behandeld.

Uitdrukkingen

Een uitdrukking wordt gevormd door combinaties van:

veldnamen
geheugenvariabelen
reekselementen
constanten
functies
operatoren
systeemgeheugenvariabelen

Geheugenvariabelen, reekselementen en functies zijn in dit hoofdstuk al eerder behandeld.

Een *veldnaam* is de naam van één *veld* of gegevensonderdeel dat zich in elk *record* van een database-bestand bevindt. Achternaam kan de veldnaam zijn van een veld waarin zich de achternamen van klanten bevinden. Kenmerkend voor elk record in dit database-bestand is dat er in het veld Achternaam één achternaam van een klant wordt ingevoerd.

Als het veld zich niet in het actieve database-bestand bevindt, moet u aan de veldnaam de aliasnaam toevoegen. U gebruikt hiervoor tussen de veldnaam en de aliasnaam het aliassymbool (->). Zoals u ziet, wordt het aliassymbool met twee toetsaanslagen ingevoerd: een koppelteken (-) en een groter-danteken (>). Voorbeeld:

```
Klant->Achternaam
```

Dit betekent: het veld Achternaam in het database-bestand waarvan de alias Klant is.

OPMERKING
Wanneer velden en geheugenvariabelen dezelfde naam hebben, krijgen velden voorrang op geheugenvariabelen. U kunt dit wijzigen door de naam van de geheugenvariabele te laten voorafgaan door het aliassymbool van geheugenvariabelen, M->.

Een *constante* is een letterlijke waarde die aan de rechter kant van de uitdrukking is ingebed, zoals "A" (een tekenconstante) of 2 (een numerieke constante).

Operatoren zijn symbolen waarmee geheugenvariabelen, velden, constanten en functies worden gekoppeld. Hierdoor kan de dBASE IV-processor de gehele uitdrukking als één eenheid berekenen. De typen operatoren worden verderop in dit hoofdstuk behandeld.

Wanneer u velden, geheugenvariabelen, constanten of een waarde die het resultaat van een functie is, in één uitdrukking combineert, moeten deze van hetzelfde *gegevenstype* zijn. Raadpleeg de bespreking van gegevenstypen in het volgende gedeelte. Converteer zonodig de elementen van niet-gelijke gegevenstypen met behulp van functies naar een gemeenschappelijk type. U moet bijvoorbeeld een numerieke variabele met een functie naar het gegevenstype teken converteren voordat u deze met tekenconstanten kunt koppelen. De uitdrukkingen in een uitdrukkingenlijst hoeven echter niet van hetzelfde gegevenstype te zijn.

Gegevenstypen

Er zijn vier gegevenstypen die in een uitdrukking kunnen worden gebruikt: *teken*, *numeriek*, *datum* en *logisch*. In werkelijkheid zijn er twee numerieke typen maar deze hoeven niet naar elkaar te worden geconverteerd. De gegevenstypen worden hierna behandeld.

De afkortingen van de dBASE IV-gegevenstypen zijn:

- <Tuitdr> voor het tekentype
- <Nuitdr> voor het binair-gecodeerde numerieke type met decimale getallen
- <Zuitdr> voor het binaire numerieke type met zwevend decimaalteken
- <Duitdr> voor het datumtype
- <voorwaarde> voor het logische type

Bovendien wordt in de syntaxis <memoveldnaam> gebruikt ter aanduiding van velden met een variabele lengte waarin grote tekstblokken worden opgeslagen. De gegevens in de memovelden worden niet in het database-bestand (.dbf) opgeslagen, maar in een afzonderlijk .dbt-bestand dat bij het .dbf-bestand hoort.

Het tekentype

Tekentype-velden, -constanten en -variabelen bevatten tekenreeksen. Tekenconstanten moeten tussen scheidingstekens staan, zoals dubbele aanhalingstekens (" "), enkele aanhalingstekens (' ') of rechte haken ([ ]). U kunt met de functie CHR() ook een decimale reeks in een tekenreeks opslaan. Voorbeelden:

```
. STORE "A" TO Mbrief
```

en

```
. STORE CHR(65) TO Mbrief
```

Met beide commandoregels wordt een tekentype-geheugenvariabele gemaakt die de letter *A* bevat.

De numerieke typen

dBASE IV kent twee numerieke gegevenstypen: *type N* en *type F*. De getallen van het type N zijn binair-gecodeerde decimale getallen. Getallen van het type F zijn de binaire getallen met een zwevend decimaalteken die in dBASE III PLUS werden gebruikt. De twee typen zijn alleen intern verschillend voor dBASE IV. Wanneer deze twee typen worden weergegeven, is er geen verschil.

Aangezien getallen van het type N een decimale notatie hebben, zijn deze niet onderhevig aan afrondingsfouten. (Er kunnen echter wel fouten optreden als de numerieke precisie wordt overschreden. Raadpleeg SET PRECISION in hoofdstuk 3 voor een bespreking van de numerieke precisie voor getallen van het type N.) Deze getallen zijn nuttig in zakelijke en financiële toepassingen waar de totalen met elkaar in evenwicht moeten zijn. Getallen van het type F hebben meer nut in wetenschappelijke toepassingen als u met erg grote of erg kleine getallen te maken hebt of als u herhaaldelijk vermenigvuldigingen of delingen uitvoert. Getallen van het type F kunnen echter door afronden of afkappen een resultaat bij benadering opleveren. Daarom zijn de getallen van het type F voor zakelijke of financiële toepassingen minder geschikt dan de getallen van het type N.

De standaardinstelling voor de getallen die u in dBASE IV invoert, is het type N. Met de functie FLOAT() kunt u deze getallen naar type F converteren. Als u echter een database-veld als type F definieert, kunt u getallen van het type F direct in dat veld invoeren zonder dat u deze met de functie FLOAT() hoeft te converteren.

Numerieke velden die u uit dBASE III PLUS importeert, worden naar getallen van het type N geconverteerd.

Als het resultaat van een functie of commando een getal is, hangt het getaltype van de functie of het commando en de invoer af. Het resultaat van de functies EXP(), LOG(), SQRT() en alle trigonometrische functies is altijd een getal van het type F. Het resultaat van alle andere functies is een getal van het type N of een getal van hetzelfde type als de invoer. Merk op dat de uitvoer van een bewerking waarin verschillende getaltypen zijn gecombineerd, een getal van het type F is.

Als de getallen erg groot of erg klein zijn, worden de getallen van zowel het type F als het type N in wetenschappelijke notatie weergegeven. De exponent wordt voorafgegaan door de letter *E*, bijvoorbeeld ,6E + 23.

Het datumtype

Met datumtype-velden en -geheugenvariabelen slaat u datums op. De grootte van een datumvariabele of datumveld is altijd acht bytes en neemt in het geheugen in totaal negen bytes in beslag. dBASE IV valideert de datumvariabelen zodra deze worden ingevoerd of gewijzigd. De standaard-datumindeling is de Nederlandse indeling, dd-mm-jj. Met SET DATE kunt u de indeling wijzigen, maar u kunt ook de instelling DATE in Config.db wijzigen. Raadpleeg *Aan de slag met dBASE IV* voor het wijzigen van de parameters van Config.db.

In dBASE IV kunt u met scheidingstekens datumwaarden aanduiden, namelijk met accolades ({ }). Deze scheidingstekens zijn gelijk aan de functie CTOD(). {12-20-59} bijvoorbeeld, is gelijk aan CTOD("12-20-59").

Een datum kan van een andere datum worden afgetrokken. Het resultaat hiervan is een getal (het aantal dagen tussen de datums). Een getal (dat een aantal dagen voorstelt) kan bij een datum worden opgeteld of van een datum worden afgetrokken. Het resultaat hiervan is eveneens een datum.

Een lege datum voert u op een van de volgende manieren in:

```
{ }
{ / / }
CTOD(" / / ")
```

Het logische type

Logische velden en logische variabelen worden als waar (.T.) of als onwaar (.F.) opgeslagen. In logische velden en logische variabelen kunnen T, t, J of j voor waar en F, f, N of n voor onwaar worden gebruikt. U moet punten als scheidingstekens gebruiken (bijvoorbeeld STORE .T. TO Mlogisch). Een logische uitdrukking wordt ook wel een *voorwaarde* genoemd.

Operatoren

dBASE IV heeft vier typen operatoren: *rekenkundige operatoren*, *relationele operatoren*, *logische operatoren* en *tekenreeks-operatoren*.

Rekenkundige operatoren

Met rekenkundige operatoren worden numerieke resultaten gegenereerd.

+	optellen/positieve monadische operator
-	aftrekken/negatieve monadische operator
*	vermenigvuldigen
/	delen
** of ^	machtsverheffen
()	haakjes voor groeperen

Relationele operatoren

Met relationele operatoren worden logische resultaten gegenereerd, met andere woorden: waar (.T.) of onwaar (.F.). Relationele operatoren kunt u in teken-uitdrukkingen, numerieke uitdrukkingen, datum-uitdrukkingen of logische uitdrukkingen gebruiken. De twee uitdrukkingen die u in een relationele bewerking gebruikt, moeten van hetzelfde type zijn. Logische vergelijkingen van tekenreeksen worden beïnvloed door het commando SET EXACT.

<	kleiner dan
>	groter dan
=	gelijk aan
< > of #	niet gelijk aan
<= of =<	kleiner dan of gelijk aan
>= of =>	groter dan of gelijk aan
$	subtekenreeks-vergelijking (als bijvoorbeeld A en B tekenreeksen zijn, resulteert A$B in logisch waar (.T.) als A identiek aan B is of als A onderdeel van B uitmaakt).

Logische operatoren

Met logische operatoren wordt uit de vergelijking van twee uitdrukkingen een logisch resultaat verkregen.

.AND.	logisch en
.OR.	logisch of
.NOT.	logisch niet
()	haakjes voor groeperen
=	gelijk aan
< > of #	niet gelijk aan

Tekenreeks-operatoren

Met tekenreeks-operatoren worden twee of meer tekenreeksen tot één tekenreeks aaneengeschakeld.

+	de spaties tussen de tekenreeksen blijven behouden wanneer de tekenreeksen worden gekoppeld
-	de spaties achter de tekenreeks die voorafgaat aan de operator worden naar het einde van de laatste tekenreeks verplaatst
()	haakjes voor groeperen

Prioriteit van de operatoren

Elk type operator is onderworpen aan een set met regels die de volgorde bepalen waarin de bewerkingen worden uitgevoerd. Deze regels worden de *prioriteitsvolgorde* van de operator genoemd.

Relationele operatoren en tekenreeks-operatoren hebben slecht één prioriteitsniveau. Deze operatoren worden in volgorde van links naar rechts uitgevoerd.

Rekenkundige operatoren

De prioteitsniveaus van rekenkundige operatoren zijn:

1. positieve (+) en negatieve (-) monadische operatoren
2. machtsverheffen
3. vermenigvuldigen en delen
4. optellen en aftrekken

Logische operatoren

De prioriteitsniveaus van logische operatoren zijn:

1. .NOT.
2. .AND.
3. .OR.

Combinaties van operatoren

Wanneer de vier verschillende typen operatoren in dezelfde uitdrukking worden gebruikt, zijn de prioriteitsniveaus als volgt:

1. rekenkundige operatoren of tekenreeks-operatoren
2. relationele operatoren
3. logische operatoren

Alle bewerkingen op hetzelfde prioriteitsniveau worden in volgorde van links naar rechts uitgevoerd. De volgorde waarin de bewerkingen worden uitgevoerd, kunt u met haakjes wijzigen. De bewerkingen die zich in de binnenste geneste haakjes bevinden, worden het eerst uitgevoerd.

SET-commando's gebruiken

Met de SET-commando's bestuurt u de systeemparameters van dBASE IV vanaf de commandostip of vanuit programmabestanden en procedurebestanden. In hoofdstuk 3 worden alle SET-commando's van dBASE IV en de bijbehorende standaardinstellingen in alfabetische volgorde behandeld. Een aantal van deze commando's wordt bij het opstarten van dBASE IV met standaardinstellingen geïnstalleerd.

Vanaf de commandostip kunt u alle SET-commando's tijdelijk wijzigen. Deze instellingen zijn echter alleen geldig totdat u ze met een ander SET-commando opnieuw instelt of totdat u het commando QUIT gebruikt. Een aantal standaardinstellingen die bij het opstarten van dBASE IV worden geactiveerd, kunt u wijzigen door het bestand Config.db aan te passen. Raadpleeg *Aan de slag met dBASE IV* voor informatie over de instellingen van Config.db.

De syntaxis van de SET-commando's kan op twee veel gebruikte manieren worden geschreven:

```
SET <parameter> ON/OFF
```

of

```
SET <parameter> TO <uitdrukking>
```

Sommige SET-commando's van dBASE III PLUS, zoals SET DOHISTORY, SET FIXED en SET MENU, worden niet gebruikt in dBASE IV. Vanwege de compatibiliteit met programma's van dBASE III PLUS verschijnt er in dBASE IV geen foutmelding wanneer een van deze SET-commando's zich in een programma bevindt of vanaf de commandostip wordt ingevoerd. Deze SET-commando's hebben geen effect omdat in het nieuwe ontwerp van dBASE IV geen behoefte meer is aan deze instellingen.

Functies gebruiken

Met behulp van functies kunt u gespecialiseerde bewerkingen uitvoeren, waardoor de commando's van dBASE IV nog meer mogelijkheden krijgen. Functies leveren een waarde op. Sommige functies toetsen of converteren gegevens en sommige voeren een handeling uit.

Achter de namen van alle dBASE IV-functies staan haakjes, behalve achter de functie voor macrosubstitutie die in het volgende gedeelte wordt behandeld. Tussen de haakjes kunnen wel of geen parameters staan die moeten worden berekend.

Met de functie voor macrosubstitutie (&) geeft u dBASE IV opdracht de *inhoud* van een tekengeheugenvariabele op te halen en niet de naam van de geheugenvariabele. Deze functie gebruikt u door voor de naam van de geheugenvariabele een en-teken (&) te plaatsen.

```
. STORE "Achternaam, Voornaam" TO Mvelden
. LIST &Mvelden
```

In dit voorbeeld is LIST &Mvelden gelijk aan LIST Achternaam, Voornaam.

Met de functie voor macrosubstitutie kunt u de gebruiker door middel van een aanwijzing gegevens laten invoeren en die gegevens onmiddellijk als onderdeel van het commando gebruiken. Met macrosubstitutie kunt u een gedeelte van een commando samenstellen waaraan de gebruiker bepaalde argumenten kan toevoegen.

Wanneer u een macro gebruikt, wordt de commandoregel door dBASE IV niet tijdens de compilatie gecompileerd. Wanneer het commando tijdens de uitvoering wordt aangetroffen, wordt het uitgebreid met de actieve waarde van de geheugenvariabele die volgt op het en-teken (&), en moet de regel opnieuw worden gecompileerd. Hierdoor wordt de verwerkingstijd langer.

In dBASE IV kunt u een macro met een variabele in plaats van een commando gebruiken. Bijvoorbeeld:

```
. Commando = [RESET IN 1]
. &Commando
```

In een dergelijk geval moet dBASE IV het commando natuurlijk steeds compileren wanneer het wordt uitgevoerd. Afhankelijk van het commando kan dit betekenen dat de volledige compiler weer moet worden opgeroepen.

In RunTime-toepassingen kunt u echter geen macro's in plaats van commando's gebruiken.

Als u de macro-uitbreiding alleen in uitdrukkingen gebruikt, kunnen de meeste commando's tijdens de compilatie worden gecompileerd en is het niet nodig de compiler weer tijdens de uitvoering op te roepen.

TIP
De verwerkingstijd kunt u korter maken door een indirecte verwijzing naar een bestandsnaam te gebruiken in plaats van macrosubstitutie (indirecte verwijzingen zijn in dit hoofdstuk al in het gedeelte "Bestandsnamen" behandeld).

Door de gebruiker gedefinieerde functies

U kunt in dBASE IV uw eigen functies definiëren. Dergelijke door u zelf gedefinieerde functies zijn een uitbreiding op de taal van dBASE IV en zorgen ervoor dat u nog meer bewerkingen kunt uitvoeren.

Wat is een door de gebruiker gedefinieerde functie?

Een UDF is een speciale procedure die u vanaf de commandoregel van dBASE IV kunt aanroepen. De procedure begint met het commando FUNCTION en bestaat uit commando's en een lijst met optionele parameters die de functie gebruikt om een waarde op te leveren. Vervolgens geeft u deze UDF een naam. Het commando FUNCTION herkent de UDF als een speciaal soort procedure. In tegenstelling tot andere procedures moet u de UDF afsluiten met het commando RETURN en de resulterende waarden. Raadpleeg voor meer informatie over het gebruik van FUNCTION en RETURN de betreffende gedeelten in hoofdstuk 2, "Commando's".

Nadat u een UDF hebt gemaakt kunt u deze op dezelfde manier als elke andere functie van dBASE IV vanaf de commandoregel aanroepen. De syntaxis die gebruikt wordt om een UDF aan te roepen is als volgt:

<UDF-naam>([<parameterlijst>])

Wanneer dBASE IV de naam van een UDF op de commandoregel aantreft, wordt de procedure van die UDF gezocht en vervolgens uitgevoerd. Daarom moeten de UDF op de commandoregel en het gerelateerde commando FUNCTION uit de procedure beslist dezelfde naam hebben. De lijst met optionele parameters bevat uitdrukkingen die naar de UDF worden overgebracht. In de UDF kan een commandoregel PARAMETERS staan waarmee lokale variabelen worden toegewezen aan elk van de onderdelen van de lijst met parameters. De haakjes [()] die achter de UDF-naam zijn geplaatst zijn verplicht, ook als er geen parameters worden opgegeven.

Een voorbeeld van een UDF

In het volgende hypothetische voorbeeldprogramma met de naam Kalender worden de essentiële onderdelen van een UDF toegelicht.

In het eerste deel van het programma wordt de gebruiker gevraagd een jaartal in te voeren. Het programma geeft datums en andere informatie weer die in dit voorbeeld niet zijn opgenomen. Voor 29 februari hoeft het programma alleen maar tekst weer te geven als het een schrikkeljaar betreft. In alle andere gevallen moeten op de plaats van 29 februari spaties komen te staan. De commandoregel

@ 19,45 SAY IIF(SCHRIK(jaar), "29 februari", SPACE (11))

geeft ofwel de tekens voor 29 februari weer, ofwel spaties, afhankelijk van de waarde die de UDF met de naam SCHRIK() oplevert. Aan dit commando is een parameter (jaar) toegevoegd, waarmee het jaartal wordt bedoeld dat de gebruiker in de UDF heeft ingevoerd. Als de functie waar (.T.) is, worden de tekens "29 februari" weergegeven. Als de UDF onwaar (.F.) IS, staan er 11 spaties en worden er geen tekens weergegeven. Raadpleeg hoofdstuk 4, "Functies", voor meer informatie over het gebruik van de functie IFF().

De UDF waarmee wordt vastgesteld of het een schrikkeljaar betreft of niet begint met het commando FUNCTION en wordt afgesloten met het commando RETURN. De procedure is in dit geval opgenomen in het bestand Kalender.prg, maar kan even goed van elk ander procedurebestand deel uit maken. De naam van de UDF, SCHRIK(), komt overeen met de aanroepopdracht uit de hiervoor genoemde @-commandoregel. Met de procedure wordt een lokale variabele jr gemaakt voor de parameter (jaar) die in de @-commandoregel is opgenomen. Bij de volgende regel wordt .T. opgeslagen voor de variabele isschrik wanneer het een schrikkeljaar betreft, en .F. als dat niet het geval is. Raadpleeg de functie MOD() in hoofdstuk 4, "Functies", voor meer informatie over het gebruik van dit commando. De commandoregel RETURN zendt .T. of .F. terug naar de commandoregel waarmee de UDF is aangeroepen.

```
* Kalender.prg
*
SET TALK OFF
jaar = 0
@ 12, 12 SAY "Voer een nieuw jaartal in: " GET jaar PICTURE "9999"
READ
.
.
.
@ 19,45 SAY IIF(SCHRIK(jaar), "29 februari", SPACE (11))
.
.
.
FUNCTION SCHRIK
PARAMETERS jr
isschrik = ((MOD(jr,4) = 0) .AND. (MOD(jr,100) <> 0)) .OR. (MOD(jr,400) = 0)
RETURN isschrik
```

Beperkingen bij het gebruik van UDF's

De manier waarop u UDF's gebruikt, wordt op twee manieren beperkt: door algemene regels met betrekking tot de UDF en door de inherente beperkingen van de tussenliggende commando's waarmee u werkt. De algemene regels zijn van toepassing bij het maken of uitvoeren van een UDF. Met de beperkingen van de tussenliggende commando's krijgt u te maken bij indelingsbestanden, de verwerking van ON-commando's en bij de UDF zelf.

Algemene regels met betrekking tot UDF's

De naam van de UDF mag niet die van bestaande dBASE IV-commando's of -functies zijn. Bij gelijke namen zal de dBASE IV-functie worden uitgevoerd en niet uw UDF. U kunt een UDF om die reden bijvoorbeeld nooit EXP noemen, want EXP() is een bestaande functie in dBASE IV.

U kunt UDF's zodanig dat een UDF de volgende UDF aanroept. Het maximum aantal UDF's daarbij is theoretisch 32, maar de grens wordt in de praktijk bepaald door de complexiteit van elk van de geneste UDF's.

De UDF mag niet worden gebruikt in een SQL-commando en andersom mag een SQL-commando niet in een UDF worden gebruikt. De UDF kan wel worden aangeroepen vanuit een SQL-programma, zolang de aanroep maar niet gebeurt door een SQL-commando en zolang de UDF zelf geen SQL-commando's bevat. Verder mag de UDF geen andere dBASE IV-commando's bevatten die in de SQL-modus verboden zijn. Raadpleeg bijlage H voor een overzicht van de dBASE IV-commando's die in de SQL-modus zijn toegestaan.

Beperkingen van tussenliggende commando's

Het gebeurt vaak dat een dBASE IV-commando door een UDF wordt onderbroken, zodat andere commando's kunnen worden uitgevoerd. De commando's die tijdens deze onderbrekingen worden uitgevoerd zijn de zogenaamde tussenliggende commando's. Ook met het commando SET FORMAT en enkele van de ON-commando's worden andere in uitvoering zijnde commando's onderbroken voor het uitvoeren van tussenliggende commando's. Alle tussenliggende commando's zijn gebonden aan bepaalde regels, ongeacht of deze door ON, SET FORMAT of door UDF's worden aangeroepen.

Zo kunt u APPEND, BROWSE en EDIT als tussenliggende commando's niet recursief gebruiken. Het is dus niet toegestaan met BROWSE door een database-bestand te bladeren als daarin een rekenveld is opgenomen waarmee een UDF wordt aangeroepen die voor hetzelfde bestand vervolgens opnieuw het commando BROWSE (of APPEND of EDIT) geeft.

Evenmin is het toegestaan LIST en DISPLAY als tussenliggende commando's recursief te gebruiken, ook niet als de records in een ander werkgebied worden uitgelijst of weergegeven. Het mag daarom niet zo zijn dat een UDF die vanuit een commando LIST wordt aangeroepen een volgend LIST- of DISPLAY-commando bevat.

Andere beperkingen van tussenliggende commando's hangen af van de instellingen van DBTRAP. DBTRAP vormt een beschermingsmechanisme tegen programmafouten, die u kunnen verrassen met ongewenste uitkomsten. Hoewel DBTRAP opnieuw kan worden ingesteld met het commando SET DBTRAP of door DBTRAP in Config.db te wijzigen, moet u de instelling ervan niet veranderen. Wilt u toch het een en ander wijzigen, overtuig u dan eerst van de mogelijke gevolgen en neem desgewenst andere beschermingsmaatregelen.

Zie hoofdstuk 3, "SET-commando's", voor meer informatie over de beperkingen en het gebruik van SET DBTRAP.

Systeemgeheugenvariabelen gebruiken

Systeemgeheugenvariabelen zijn geheugenvariabelen die door dBASE IV automatisch worden gemaakt en opgeslagen. U bepaalt hiermee het uiterlijk van de uitvoer naar de printer en het scherm en u slaat hierin printerinstellingen op. Met systeemgeheugenvariabelen bepaalt u in het bijzonder:

- de kenmerken van afdruktaken, zoals de paginadoorvoer en het aantal af te drukken exemplaren,

- het uiterlijk van alinea's, zoals de lijning, het inspringen en de marges,

- het uiterlijk van de afgedrukte pagina, zoals de tekenbreedte, de afdrukkwaliteit, de paginalengte en de linker afdrukmarge van de pagina

- de standaardinstellingen van de tekstverwerker, zoals de afstand tussen de tabinstellingen

De namen van alle systeemvariabelen beginnen met een onderstrepingsteken (_) zodat deze van gewone geheugenvariabelen kunnen worden onderscheiden (u mag geen andere geheugenvariabelen die met een onderstrepingsteken beginnen, definiëren).

Wanneer dBASE IV wordt opgestart, worden de standaardinstellingen van de systeemvariabelen automatisch geïnitialiseerd. U kunt deze waarden met het rapportontwerpscherm en het etiketontwerpscherm, vanaf de commandostip of in een programma wijzigen. Door de commando's CLEAR MEMORY en RELEASE worden de systeemgeheugenvariabelen niet uit het geheugen verwijderd.

De normale regels voor het bereik voor geheugenvariabelen zijn ook van toepassing op systeemgeheugenvariabelen. Wanneer u een programma hebt uitgevoerd dat aangepaste systeemvariabelen bevat, worden de oorspronkelijke instellingen automatisch weer teruggezet. U kunt deze variabelen niet met RELEASE opheffen.

Veel systeemgeheugenvariabelen hebben invloed op de gegevensstroom die de uitvoer vormt. Deze *gegevensstroom* wordt door alle uitvoercommando's geproduceerd, behalve door de commando's @, @...TO en EJECT. De positie van de gegevensstroom wordt niet bepaald met rij- en kolomcoördinaten. De bestemmingen van de uitgevoerde gegevensstroom kunnen echter met de commando's SET CONSOLE, SET PRINTER en SET ALTERNATE en met de optie TO PRINTER/TO FILE <bestandsnaam> van commando's zoals LIST/DISPLAY worden bestuurd.

De gegevensstroom begint bij de positie van de cursor op het scherm of bij de positie van de printkop of bij de positie van de bestandswijzer als de gegevens naar een schijfbestand worden gezonden.

OPMERKING
Hoewel de commando's @ en @...TO geen gegevensstroom als uitvoer produceren, kan door deze commando's toch de positie van de uitvoer daarna worden beïnvloed. Wanneer u bijvoorbeeld de positie van de cursor op het scherm met een @-commando wijzigt, beïnvloedt u de positie waar een volgend ?/??-commando wordt weergegeven.

De systeemgeheugenvariabelen die invloed hebben op de uitgevoerde gegevensstroom, zijn _box, _pageno, _pcolno, _pform, _plength, _plineno, _pspacing en _tabs.

Andere systeemgeheugenvariabelen die geen invloed hebben op de uitgevoerde gegevensstroom kunnen worden ingedeeld als *printer-specifiek*, *afdruktaak-specifiek* of *alinea-specifiek*.

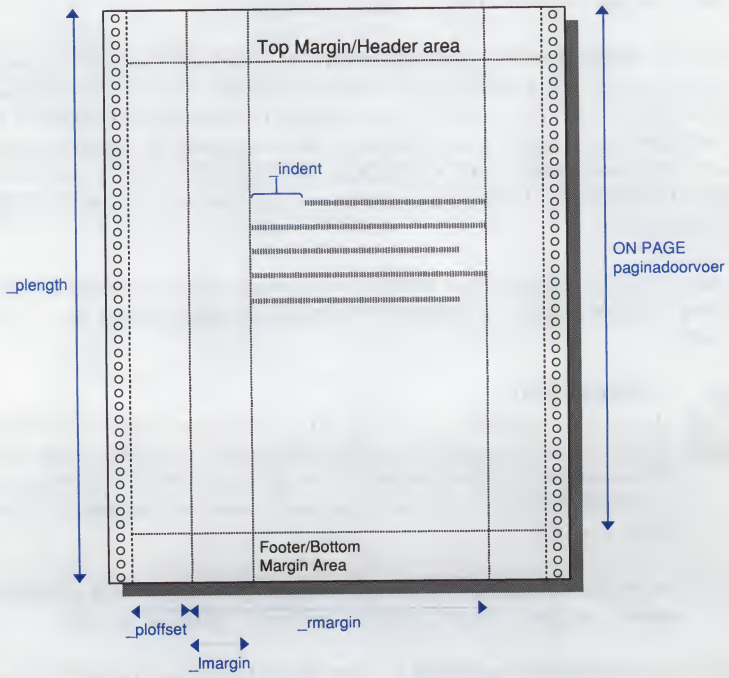
De printerinstellingen bestuurt u met de printer-specifieke systeemgeheugenvariabelen: _padvance, _pdriver, _ploffset, _ppitch, _pquality en _pwait.

Met het commando PRINTJOB activeert u de afdruktaak-specifieke systeemgeheugenvariabelen: _pbpage, _pcopies, _pecode, _peject, _pepage en _pscode.

De indeling van de tekst beïnvloedt u met de alinea-specifieke systeemgeheugenvariabelen: _alignment, _indent, _lmargin, _rmargin en _wrap.

In afbeelding 1-1 worden verschillende systeemgeheugenvariabelen getoond. Het betreft de volgende systeemgeheugenvariabelen:

- _plength; paginalengte
- _ploffset; linker afdrukmarge van de pagina
- _lmargin; linker marge vanaf _ploffset
- _rmargin; aantal kolommen van rechter marge
- _indent; inspringen van alinea



Afbeelding 1-1 Pagina-opmaak

Door de ON PAGE-procedures voor kop- en voetregeltekst wordt de af te drukken pagina ook

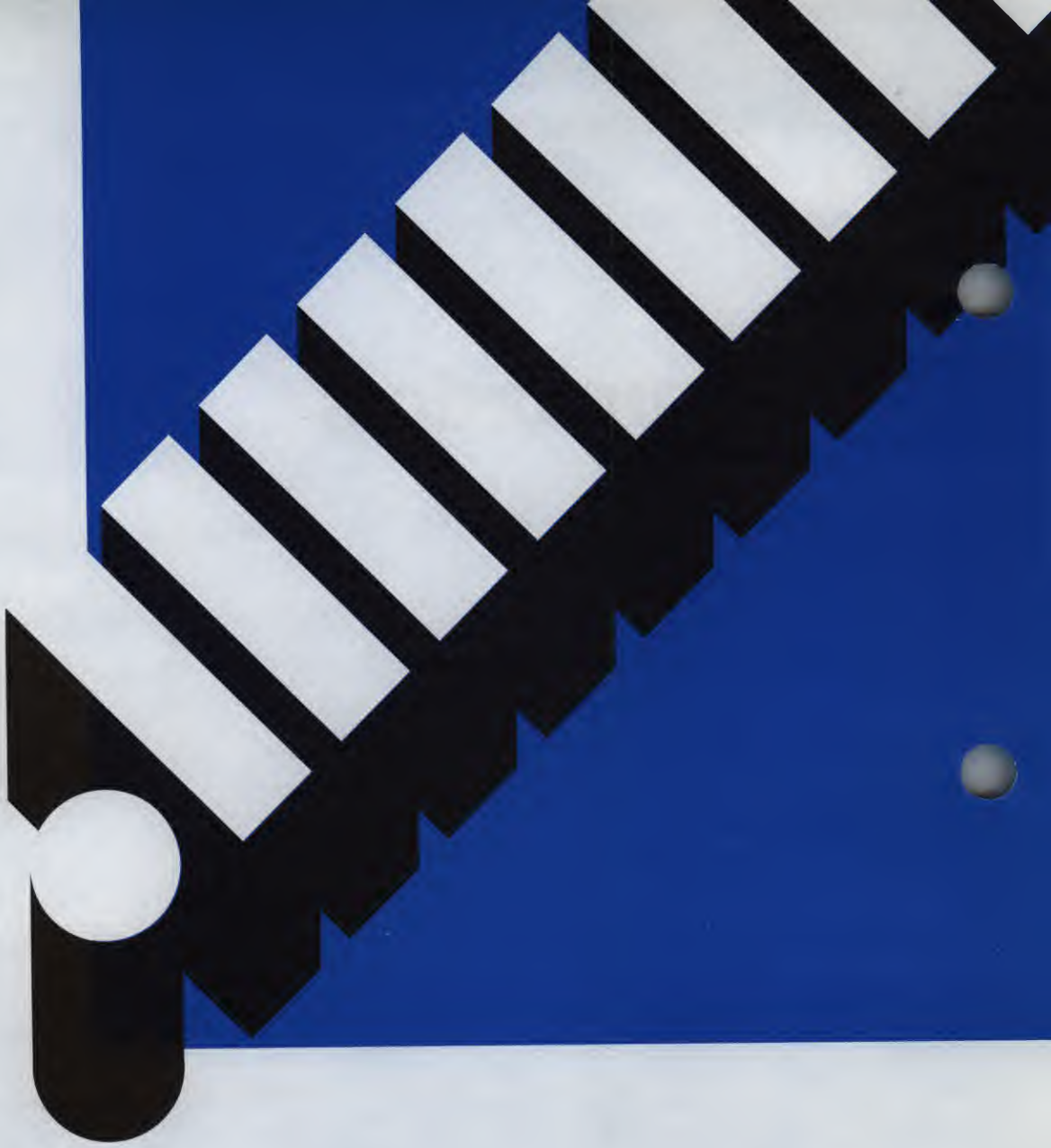
beïnvloed. Raadpleeg de bespreking van ON PAGE en de voorbeelden van paginadoorvoer in hoofdstuk 2, "Commando's".

Door het rapportontwerpscherm en het etiketontwerpscherm worden de systeemvariabelen voor u verwerkt. Wanneer u met CREATE/MODIFY REPORT een rapport of met CREATE/MODIFY LABEL een etiket maakt, kunt u de gewijzigde definities van de systeemvariabelen in een binair *afdrukmodel* opslaan, dat de extensie .prf heeft. Als u de standaardinstellingen in dit bestand wilt wijzigen, raadpleegt u *Menu's van dBASE IV*. U kunt ook nog bijbehorende, aangepaste afdrukbestanden maken. Deze bestanden hebben meestal dezelfde naam als het rapport of het etiket en de extensie .prf.

dBASE IV activeert dit afdrukmodel wanneer u daarna het rapport of het etiket wijzigt of wanneer u Kiezen model uit het menu Afdrukken kiest. Met het commando REPORT FORM wordt het afdrukmodel echter niet automatisch geactiveerd. Voordat u het commando REPORT FORM gebruikt, moet u de afdrukinstellingen opgeven. U doet dit door _pform gelijk te stellen aan de naam van het afdrukmodel of door wijzigingen aan te brengen in afzonderlijke systeemvariabelen.

Naslaghandboek van dBASE IV

Commando's



Commando's

?/??

Met ?/?? wordt de waarde van een of meer uitdrukkingen berekend en weergegeven, waarbij indeling, type en bestemming optioneel zijn. De uitvoer van ?/?? kan worden gezonden naar een scherm, een printer, een ALTERNATE-bestand of een combinatie daarvan.

Syntaxis

?/?? [<uitdrukking 1>]
[PICTURE <Tuitdr>]
[FUNCTION <functielijst>]
[AT <Nuitdr>] [STYLE <fontnummer>]]
[,<uitdrukking 2> ...] [,]

Gebruik

Wanneer SET CONSOLE op ON is ingesteld, wordt de uitvoer van de commando's ?/?? naar het scherm gezonden. Als SET PRINT op ON is ingesteld, wordt de uitvoer van de commando's ?/?? naar de printer gezonden. Als er een ALTERNATE-bestand is opgeven via de instelling van SET ALTERNATE TO <bestandsnaam> en SET ALTERNATE op ON is ingesteld, wordt de uitvoer van de commando's ?/?? naar het betreffende ALTERNATE-bestand gezonden. De instellingen voor scherm, printer en ALTERNATE-bestand kunnen afzonderlijk en in combinaties op ON zijn ingesteld.

Met het ?-commando wordt eerst een terugloop en een regeldoorvoer verzonden voordat de resultaten van de uitdrukkingenlijst worden weergegeven.

Bij het ??-commando is dit niet het geval en begint de weergave van de uitdrukkingenlijst op de positie van de cursor of printer

U kunt net zoveel uitdrukkingen en optionele PICTURE-, FUNCTION-, AT-, en STYLE-clausules invoeren als op de commandoregel van 1024 tekens passen.

Wanneer u meer uitdrukkingen wilt afbeelden dan op de commandoregel passen, kunt u met de volgkomma aangeven dat er nog een commando volgt. De volgkomma geeft aan dat met de uitvoer van alle uitdrukkingen in het actieve commando ?/?? moet worden gewacht tot het eerstvolgende commando ?/?? waarin geen volgkomma is opgenomen, is uitgevoerd. Bijvoorbeeld:

```
* Detailzone afdrukken
?? dat_trans                AT 0,
?? art_id                   AT 10,
?? Goeder->part_name        AT 21,
?? art_kwal                 AT 53 PICTURE "999",
?? Goeder->prijs            AT 58 PICTURE "99,999.99",
* Verh prijs
tot_prijs = ROUND(art_kwal * Goeder->prijs,2)
? tot_prijs                 AT 70 PICTURE "99,999.99"
?
```

Opties

Met de opties PICTURE, FUNCTION, AT en STYLE kunt u het uiterlijk van uitvoer aan uw eigen wensen aanpassen. Bij uitvoer naar een ALTERNATE-bestand wordt de STYLE-clausule genegeerd.

Met de PICTURE-templates en de FUNCTION-clausules wordt de uitvoer ingedeeld. Alle symbolen en functies van de PICTURE-templates werken met het commando ?/??. Meer informatie over de symbolen vindt u in het gedeelte over het commando @.

Drie functies (V, H en ;) werken uitsluitend met het ?/??-commando.

Met de functie V <n> wordt opgegeven dat uitdrukkingen door het commando ?/?? in verticale kolommen met een breedte van <n> tekens worden weergegeven. Als @V niet wordt gebruikt, is de standaardbreedte van memovelden gelijk aan het aantal tekens dat u met SET MEMOWIDTH hebt ingesteld. Wanneer u een veld verticaal vergroot met @V, kan de tekst in dat veld doorlopen. Alle velden behalve memovelden blijven afhankelijk van de PICTURE-template als er een is opgegeven, waardoor tekst aan de rand kan worden afgebroken als de tekenreeks langer is dan in de template is toegestaan.

Het gebruik van de functie V met de parameter 0 heeft alleen invloed op memovelden en niet op andere soorten velden. Door V(0) worden de regels van het veld in de uitvoer weergegeven zoals ze in de tekstverwerker werden weergegeven. Wanneer u velden afdrukt die met de functie V zijn vergroot, wordt nadat het afdrukken is voltooid, automatisch een regelonderbreking ingevoegd voordat met het afdrukken van het V(0)-veld wordt begonnen.

De functie H wordt bij memovelden slechts gebruikt wanneer de systeemgeheugenvariabele _wrap waar (.T.) is. Bij deze functie worden de woorden in het memoveld afgebroken volgens de instellingen van de systeemgeheugenvariabelen _lmargin en _rmargin.

Bij gebruik van de functie ; wordt tekst afgebroken op de plaats waar een puntkomma staat. De puntkomma's in de tekst verschijnen niet in de uitvoer. Raadpleeg hoofdstuk 5, "Systeemgeheugenvariabelen", voor meer gegevens over de verschillende indelingsbesturingen.

dBASE IV voert alinea-onderbrekingen in memovelden altijd uit.

Er zijn uitdrukkingen met gegevens die de opgegeven PICTURE-template niet geheel vullen. Het gaat hier om zogenaamde korte velden. Er bestaan vier functies waarmee deze korte velden binnen de PICTURE-template kunnen worden uitgelijnd.

@ B Lijnt tekst binnen een veld links uit.
@ I Centreert tekst binnen een veld.
@ J Lijnt tekst binnen een veld rechts uit.
@ T Wordt gebruikt in combinatie met uitlijnfuncties en verwijdert de voorloop- en volgspaties uit een veld.

Het commando ?/?? ondersteunt ook twee andere functies: $ en L. Met de functie $ wordt een valutateken voor of achter het bedrag weergegeven. Als SET CURRENCY op LEFT is ingesteld, wordt het valutateken voor het bedrag weergegeven. Als SET CURRENCY op RIGHT is ingesteld, wordt het valutateken achter het bedrag weergegeven. Met de functie L wordt een kort veld met voorloopnullen weergegeven.

Als een ?/??-commando eindigt met een komma, wordt de eerste regel van alle verticaal vergote velden weergegeven, maar worden de daaropvolgende regels van het verticale veld pas weergegeven wanneer het systeem een van de volgende commando's krijgt:

1. Er wordt een commando ENDPRINTJOB uitgevoerd
2. Er wordt een commando ? uitgevoerd
3. Er wordt een commando ?? zonder volgkomma uitgevoerd.

Bij 2. en 3. heeft een commando ?/?? dat wordt uitgevoerd binnen een commando ON (bijvoorbeeld ON ESCAPE, ON PAGE), niet tot gevolg dat de uitvoer van verticaal vergrote velden wordt afgebroken als die tenminste al begonnen was voordat het commando ON werd uitgevoerd. De uitvoer van dergelijke velden die is gestart tijdens de uitvoering van een commando ON, wordt door een commando zoals bedoeld bij 2. en 3. wel afgebroken.

Met de optionele AT-clausule kunt u opgeven in welke kolom de uitdrukking wordt weergegeven. De numerieke uitdrukking moet liggen tussen 0 en 255. U kunt met deze optie tekstkolommen naast elkaar afdrukken, ongeacht de lengte van de afgedrukte tekst links van de kolom.

Met de optionele STYLE-clausule kunt u verschillende schriftsoorten opgeven, bijvoorbeeld vet of cursief. Of met STYLE de weergave op het scherm wordt gewijzigd is afhankelijk van het beeldscherm waarmee u werkt. STYLE beïnvloedt echter wel de manier waarop de uitvoer wordt afgedrukt. De STYLE-clausule kan bestaan uit letters, cijfers of een combinatie hiervan.

De toegestane letters zijn:

B	-	vet
I	-	cursief
U	-	onderstreept
R	-	verhoogd (superscript)
L	-	verlaagd (subscript)

De toegestane cijfers zijn 1 tot en met 5. De cijfers activeren de fonts die u eerder hebt gedefinieerd. De besturingscodes voor begin en einde van de fonts zijn vastgelegd in Config.db via de opdracht PRINTER <printer int> LETTERTYPE <lettertype int>.

U kunt verschillende schriftsoorten combineren en meerdere schriftsoorten op dezelfde regel afdrukken.

U kunt ook het commando ??? en de systeemgeheugenvariabelen _pscode en _pecode gebruiken om het lettertype te wijzigen. Normaal gesproken wijzigt u het lettertype van afzonderlijke tekstdelen met het commando ?/?? en de optie STYLE. Met het commando ??? wijzigt u het lettertype in grotere tekstdelen. Met _pscode en _pecode kunt u het lettertype voor een geheel document definiëren.

Tips

Als u ? zonder uitdrukking opgeeft, wordt een blanco regel weergegeven. Deze optie kunt u gebruiken om in de uitvoer een regel over te slaan. Als u echter enkele of dubbele regelafstand voor de uitvoer wilt instellen, gebruikt u de systeemgeheugenvariabele _pspacing. U kunt het commando ?? zonder uitdrukking gebruiken om een functie buiten werking te stellen (bijvoorbeeld: ON KEY LABEL F2 ??, waarna de functietoets F2 buiten werking is).

Speciaal geval

Met de functies B, I of J kunt u de gehele instelling van _alignment voor een memoveld of de uitlijnopties in afzonderlijke alinea's laten vervallen. Met de volgende routine centreert u een memoveld met de naam Aantekeningen verticaal en vergroot u het veld binnen de 15 tekens brede weergavekolom die door FUNCTION is gedefinieerd. De verticale centrering heeft prioriteit boven de oorspronkelijke instelling (_alignment = "LEFT").

```
_alignment = "LEFT"
_wrap = .T.
? Aantekeningen FUNCTION "IV15"
```

Als _wrap op onwaar (.F.) is ingesteld, wordt de functie H bij het afdrukken van een memoveld genegeerd.

Voorbeelden

Van het database-bestand Afnemers moeten de waarden van de velden Klant_nr en Klantnaam vet worden afgedrukt en de waarde van Achternaam cursief:

```
. USE Afnemers<
. ET PRINT ON
. ? Klant_nr STYLE "B", Klantnaam STYLE "B", Achternaam STYLE "I"
A00001 WIJSMAN & ZONEN BV        Wijsman
```

Het doorhalen van tekst is soms nuttig om wijzigingen in een document aan te geven. Om een regel tekst uit een programmabestand door te halen, gebruikt u de optie AT en stelt u _wrap in op onwaar (.F.). (In hoofdstuk 5 leest u meer over _wrap en de andere systeemgeheugenvariabelen.)

```
_wrap=.F.
SET PRINT ON
? "In afwachting van ontvangst."
?? "////////////////////////////////////Ontvangen 7-3-88" AT 0
```

Om tekst te overschrijven in plaats van door te halen, gebruikt u bovenstaande methode, waarbij _wrap moet zijn ingesteld op waar. Bij overschrijven wordt alleen de tweede regel weergegeven, bij doorhalen worden beide regels weergegeven. Beide procedures zijn alleen van toepassing bij het maken van een afdruk: op het scherm kan een document niet worden doorgehaald of overschreven.

Zie ook

???, @, PRINTJOB, SET ALTERNATE, SET CONSOLE, SET MEMOWIDTH, SET PRINTER

Hoofdstuk 5, "Systeemgeheugenvariabelen"

"Aan de slag met dBASE IV"

???

Met het commando ??? verzendt u de uitvoer rechtstreeks naar de printer, zonder het geïnstalleerde printerstuurprogramma te gebruiken.

Syntaxis

??? <Tuitdr>

Gebruik

Met het commando ??? kunt u tekens naar de printer zenden waarmee de actieve kolom- en rijpositie van de printer ongewijzigd blijven. Meestal zendt u printerstuurcodes wanneer het printerstuurprogramma een bepaalde functie van de printer niet ondersteunt.

Het commando ?/?? en de systeemgeheugenvariabelen _pscode en _pecode zenden ook printerstuurcodes naar de printer. Normaal gesproken geeft u het commando ?/?? (STYLE-optie) om de schriftsoorten van afzonderlijke tekstdelen te wijzigen, ??? om de schriftsoorten in een groter gedeelte van de tekst te wijzigen en _pscode en _pecode om de schriftsoort voor een geheel document te definiëren.

De printerstuurcodes zijn specifiek voor de printer die u gebruikt. Raadpleeg het boek bij uw printer voor informatie over de benodigde stuurcodes.

Printerstuurcodes kunnen alle tekens bevatten die kunnen worden afgedrukt, behalve het dubbele aanhalingsteken (“). Daarnaast kunnen ook tekens voorkomen die niet kunnen worden afgedrukt, zoals Esc. U kunt deze niet-afdrukbare tekens op verschillende manieren definiëren.

Stuurcodes zijn tekenreeksen die niet-afdrukbare tekens aanduiden. De accolades ({ }) zijn verplicht en de gehele tekenreeks moet tussen aanhalingstekens worden geplaatst (zie Voorbeelden).

Tabel 2-1 Stuurcodes

ASCII-code	Stuurcode
0	{NULL} of {CTRL-@}
1	{CTRL-A}
2	{CTRL-B}
3	{CTRL-C}
4	{CTRL-D}
5	{CTRL-E}
6	{CTRL-F}

(*wordt vervolgd*)

Tabel 2-1 Stuurcodes (*vervolg*)

ASCII-code	Stuurcode
7	{DEL} of {DELETE}
8	{CTRL-H}
9	{TAB} of {CTRL-I}
10	{LINEFEED} of {CTRL-J}
11	{CTRL-K}
12	{CTRL-L}
13	{RETURN} of {CTRL-M}
14	{CTRL-N}
15	{CTRL-O}
16	{CTRL-P}
17	{CTRL-Q}
18	{CTRL-R}
19	{CTRL-S}
20	{CTRL-T}
21	{CTRL-U}
22	{CTRL-V}
23	{CTRL-W}
24	{CTRL-X}
25	{CTRL-Y}
26	{CTRL-Z}
27	{ESC} of {ESCAPE} of {CTRL-[}
28	{CTRL-\}
29	{CTRL-]}
30	{CTRL-^}
31	{CTRL-_}
127	{BACKSPACE}
403	{CTRL-_}

Tips

Als u een accolade wilt afdrukken, plaatst u deze tussen accolades, bijvoorbeeld ???"{{}". Opdrachten zoals ???CHR(123) of ???"{" zijn niet mogelijk.

Speciaal geval

Hoewel het gemakkelijker is vanuit het Control Center of vanuit een programma de uitvoer naar een PostScript-printer te zenden, kunt u ook vanaf de commandostip op een PostScript-printer afdrukken. De procedure hiervoor is als volgt:

1. Open met het commando USE het database-bestand dat u wilt afdrukken.
2. Laad het printerstuurprogramma, bijvoorbeeld:

```
. _pdriver="postscri.pr3"
. LIST TO PRINT
. ??? "FF "
```

3. Zend enige uitvoer naar de printer zodat het DLD-bestand wordt geïnitialiseerd.
4. Zend, nadat het database-bestand is afgedrukt, met de tekenreeks "FF " een doorvoercommando naar de printer zodat de laatste pagina met uitvoer wordt doorgevoerd.

Als u geen uitvoer naar de printer zendt nadat het DLD-bestand is geïnitialiseerd, wordt het bestand uit het printergeheugen verwijderd. U kunt het DLD-bestand als volgt zelf verwijderen:

```
. ??? CHR(4)
```

Met het commando ??? kunt u vanaf de computer fonts in de printer laden. Proportionele fonts worden zonder een programma echter niet op de juiste manier uitgelijnd. U kunt bijvoorbeeld het standaardfont Courier met vaste tekenbreedte als volgt naar de printer zenden:

```
. ??? "1FONT"
```

Voorbeelden

In dit voorbeeld zendt u de code Esc-E naar een Hewlett-Packard LaserJet printer (met deze code wordt de printer opnieuw ingesteld). Omdat de ASCII-code voor Esc 27 is en de letter E de ASCII-code 69 is, kunt u de volgende dingen doen:

De functie CHR() gebruiken waarmee een getal wordt omgezet in het bijbehorende ASCII-teken:

```
. ??? CHR(27) + "E"
```

Stuurcodes gebruiken:

```
. ??? "{ESC}E"
```

Alleen ASCII-codes gebruiken, waarbij de codes tussen accolades moeten staan:

```
. ??? "{27}{69}"
```

Een combinatie van ASCII-codes en letters gebruiken:

```
. ??? "{27}E"
```

Zie ook

?/??, @, CHR(), INKEY(), LASTKEY(), READKEY()

Hoofdstuk 5, "Systeemgeheugenvariabelen"

@

Hoofdstuk 2 van Aan de slag met dBASE IV

Met het commando @ kunt u zelf formulieren ontwerpen voor de invoer en uitvoer van gegevens. De gegevens worden met dit commando volgens opgegeven instellingen op de aangegeven coördinaten ingedeeld.

Syntaxis

```
@ <rij>, <kol>
    [SAY <uitdrukking>
        [PICTURE <Tuitdr>] [FUNCTION <functielijst>]]
    [GET <variabele>
        [[OPEN] WINDOW <naam venster>]
        [PICTURE <Tuitdr>]
        [FUNCTION <functielijst>]
        [RANGE [<laag>] [,<hoog>]]
        [VALID <voorwaarde>[ERROR <Tuitdr>]]
        [WHEN <voorwaarde>]
        [DEFAULT <uitdrukking>]
        [MESSAGE <Tuitdr>]]
    [COLOR [<normaal>] [,<speciaal>]]
```

Gebruik

<rij> en <kol> zijn numerieke uitdrukkingen. Als u het commando SET DEVICE TO SCREEN hebt gegeven, heeft <rij> een bereik van 0 tot de hoogte van het scherm, in regels. <kol> kan tussen 0 en 79 liggen. De coördinaten van <rij> en <kol> worden altijd berekend vanuit de linker bovenhoek van het actieve venster, of dit nu het gehele scherm is of een venster dat u zelf hebt gedefinieerd. Als u het commando SET DEVICE TO PRINTER of SET DEVICE TO FILE hebt gegeven, kan <rij> tussen 0 en 32.767 liggen en <kol> tussen 0 en 255.

Met het commando SET STATUS ON wordt regel 22 van het scherm gereserveerd voor de statusregel. Als u het commando SET STATUS OFF gebruikt maar niet het commando SET SCOREBOARD OFF, wordt regel 0 gebruikt voor het weergeven van statusinformatie. Om deze regels te kunnen gebruiken voor andere informatie, zet u SET STATUS en SET SCOREBOARD op OFF.

OPMERKING
Met @ wordt geen gegevensstroom geproduceerd, maar de positie van de daarop volgende uitvoer kan door dit commando echter wel worden beïnvloed. Als u bijvoorbeeld de positie van de cursor op het scherm met het commando @ wijzigt, heeft dit invloed op de positie waarop een later ?/??-commando wordt weergegeven.

De berekeningen worden bijgewerkt als u de inhoud van een veld wijzigt waardoor de uitkomst van andere berekeningen wordt gewijzigd of waardoor standaardinstellingen van velden die in het formulier zijn gedefinieerd, worden beïnvloed. De uitkomsten worden daarna opnieuw weergegeven zodra u op een van de stuurtoetsen drukt, bijvoorbeeld **PgUp** of **PgDn**.

Met het sleutelwoord SAY kan informatie worden weergegeven die u niet wilt wijzigen. De waarde van elke geldige dBASE IV-uitdrukking kan worden weergegeven.

Met het sleutelwoord GET kunt u de gegevens weergeven en bewerken die zijn opgeslagen in velden of die zijn toegewezen aan geheugenvariabelen of reeksen. Het commando READ activeert de GETs en maakt invoer en bewerking van gegevens op het gehele scherm mogelijk.

Naast het gebruiken van GETs in het commando READ kunt u ook een indelingsbestand (.fmt) maken dat @-commando's bevat (indelingsbestanden zijn tekstbestanden die kunnen worden gemaakt met MODIFY COMMAND of met een andere tekstverwerker). Ook met CREATE/MODIFY SCREEN kunt u zo'n invoer- of uitvoerscherm opmaken. Met dit commando wordt een .fmt-bestand gemaakt waarin @-commando's zijn opgenomen. U kunt een indelingsbestand gebruiken met de commando's APPEND, BROWSE [FORMAT], CHANGE, EDIT, INSERT en READ.

Met behulp van SET DEVICE TO PRINT kunt u @-commando's naar de printer zenden. GETs worden niet naar de printer gezonden. De meeste printers hebben beperkingen ten aanzien van rij- en kolomcoördinaten. Wanneer u @-commando's naar een printer zendt en het rijnummer in opeenvolgende @-commando's verlaagt, wordt de pagina doorgevoerd. Als twee @-commando's dezelfde rijcoördinaat hebben, moet het tweede commando daarom een hogere kolomcoördinaat hebben.

Met een door de gebruiker gedefinieerde functie kunnen waarden in de GET-variabele worden gewijzigd.

Opties

De opties van het @-commando worden in alfabetische volgorde beschreven.

@ <rij>, <kol> - Zonder opties wordt de opgegeven rij vanaf de opgegeven kolompositie gewist. De actieve waarden van <rij> en <kol> kunnen worden vervangen door de functie $. U kunt bijvoorbeeld als volgt een melding weergeven over opeenvolgende schermrijen:

```
. @ $+1,2 SAY "MELDING"
```

COLOR - Hiermee geeft u de kleuren op die worden gebruikt voor de SAY- en GET-variabelen. De kleur <normaal> wordt gebruikt voor SAYs, de kleur <speciaal> voor GETs. Voor deze kleuren gelden dezelfde regels als voor de kleuren die u instelt met het commando SET COLOR. U kunt voor SAYs en GETs een voorgrond- en een achtergrondkleur opgeven. Raadpleeg het commando SET COLOR voor meer informatie over de wijze waarop met deze codes de kleuren van het beeldscherm kunnen worden gewijzigd.

U kunt de kleuren "normaal" of "speciaal" ongewijzigd laten door een kleurcode weg te laten. Als u de code voor normaal weglaat, maar de speciale kleuren wilt wijzigen, typt u een komma vóór de code voor speciale kleuren. Op de volgende manier kunt u bijvoorbeeld de speciale kleuren wijzigen in witte tekens op een rode achtergrond:

```
. @ 2,20 GET Tekst COLOR ,W/R
```

De kleuren die u hier opgeeft, krijgen een hogere prioriteit dan het commando SET COLOR, maar alleen voor de uitvoer van het actieve @-commando.

DEFAULT - Deze uitdrukking kan alleen bij indelingsbestanden (.fmt) worden gebruikt om een vooraf ingestelde waarde in een GET-variabele op te nemen. De waarde moet overeenkomen met het gegevenstype van de GET-variabele. De uitdrukking wordt alleen verwerkt wanneer u records toevoegt aan een database-bestand. De uitdrukking DEFAULT verschijnt in de GET-variabele en wanneer u op ø drukt, wordt de waarde toegewezen aan de GET-variabele. DEFAULT-waarden worden genegeerd als met het SET CARRY-commando waarden zijn ingesteld.

ERROR - <Tuitdr> kan iedere geldige tekenuitdrukking zijn. Met deze optie kunt u een door uzelf opgegeven melding weergeven als niet wordt voldaan aan de VALID <voorwaarde>. De melding die u zelf opgeeft, verschijnt in plaats van de melding van dBASE IV: **Niet voldaan aan bewerkingsvoorwaarde**.

FUNCTION - De optie FUNCTION lijkt veel op de optie PICTURE. Raadpleeg de bespreking bij PICTURE verderop. FUNCTION heeft betrekking op het gehele onderdeel en PICTURE alleen op een opgegeven deel van het gegevensonderdeel.

MESSAGE - <Tuitdr> moet een geldige tekenuitdrukking zijn. Deze tekenuitdrukking verschijnt als een READ-commando wordt uitgevoerd en de cursor in het GET-veld staat dat bij de melding hoort. De melding verschijnt gecentreerd onder aan het scherm, tenzij u deze met de AT-clausule van het commando SET MESSAGE een andere positie geeft.

WINDOW <naam venster> - Wanneer u het @-commando gebruikt in combinatie met GET <variabele> en de variabele een memoveld is, kunt u met de optie WINDOW een nieuw bewerkvenster openen. Plaats de cursor in het memoveld en druk op **Ctrl-Home** om het venster te openen. Met **Ctrl-End** kunt u het venster weer sluiten. Als het veld tekst bevat, ziet u het woord "memo" in hoofdletters. Als het veld leeg is, staat "memo" in kleine letters.

De parameters <rij>, <kol> van @...GET die bij de optie WINDOW worden gebruikt, verwijzen naar de lokale coördinaten binnen de vensterrand.

<naam venster> is de naam van een venster dat u eerder hebt gedefinieerd met DEFINE WINDOW.

Als u de optie WINDOW niet hebt opgegeven, gebruikt dBASE IV het gehele beeldscherm. In het venster wordt de tekstverwerker gebruikt die u hebt opgegeven in het bestand Config.db.

OPEN WINDOW <naam venster> - Het bewerkvenster voor het memoveld kan standaard geopend zijn. Als u OPEN opgeeft bij de optie WINDOW, hoeft u het venster niet telkens te openen en te sluiten. U moet wel met **Ctrl-Home** en **Ctrl-End** van en naar het venster te gaan.

PICTURE - Met deze optie kunt u opgeven welk type gegevens in een variabele mogen worden opgenomen. U kunt ook bepalen hoe de gegevens moeten worden weergegeven. De clausule kan bestaan uit een functie, die wordt voorafgegaan door een @-symbool en die alle in- en uitvoertekens beïnvloedt, of een template, dat afzonderlijke in- en uitvoertekens beïnvloedt. Een PICTURE-clausule kan elke tekenuitdrukking zijn, maar is meestal een tekenreeks die tussen aanhalingstekens staat. Als de clausule een geheugenvariabele is, staat deze tussen haakjes.

De optie PICTURE kan ook variabelen die symbolen van veldopmaak-functies en templates bevatten, als tekenuitdrukking gebruiken. De indeling van variabelen verschilt niet van de indeling van symbolen van veldopmaak-functies en templates die in een programma of commandoregel worden gebruikt.

Veldopmaak-functies en templates van PICTURE worden verderop uitvoeriger beschreven. U kunt functies ook gebruiken met de optie FUNCTION. In dat geval hoeven de functies niet te worden voorafgegaan door het @-symbool. De functies PICTURE en FUNCTION kunnen op iedere commandoregel tweemaal worden gebruikt, eenmaal met SAY en eenmaal met GET.

RANGE - Met deze optie worden met numerieke variabelen, tekenvariabelen en datumvariabelen onder- en bovengrenzen opgegeven. De uitdrukkingen <laag> en <hoog> moeten van hetzelfde gegevenstype zijn. Met deze uitdrukkingen geeft u de minimum- en maximumwaarden (met inbegrip van de opgegeven waarden) op die mogen worden ingevoerd als antwoord op een GET-commando.

Als u alleen de boven- of ondergrens wilt opgeven, voert u slechts één uitdrukking in. In de uitdrukking moet u een komma (,) opnemen, bijvoorbeeld RANGE ,30 of RANGE 10, . Met de komma wordt aangegeven of het gaat om de waarde <hoog> of <laag>. Telkens wanneer u een ongeldig getal of een ongeldige datum invoert, wordt u gevraagd een nieuwe waarde in te voeren, totdat u een geldig getal of een geldige datum hebt ingevoerd.

Als u een waarde probeert in te voeren die buiten de ondergrens en de bovengrens valt, verschijnt de aanwijzing **RANGE is <ondergrens> naar <bovengrens> (druk op de SPATIEBALK).** U moet dan op de spatiebalk drukken en de waarde opnieuw invoeren. Als u in de RANGE-optie geen ondergrens en bovengrens hebt opgegeven, worden <ondergrens> en <bovengrens> in deze melding vervangen door het woord "Geen". De aanwijzing **RANGE is 1 naar Geen (druk op de SPATIEBALK)**, bijvoorbeeld, verschijnt als u in de RANGE-optie geen bovengrens hebt opgegeven en u probeert een 0 in te voeren. U moet dan op de **spatiebalk** drukken en een waarde invoeren die groter is dan 0.

Als u een RANGE opgeeft en zonder invoer op ↵ drukt als antwoord op een GET-commando, wordt het bereik niet gecontroleerd. Als een veld of een variabele een waarde heeft die buiten het opgegeven gebied valt, kunt u deze waarde dus ongewijzigd laten door op ↵ te drukken.

Met het commando ON READERROR en met alle commando's die hierdoor kunnen worden uitgevoerd, wordt voorkomen dat de meldingen bij het controleren van het bereik worden weergegeven.

VALID - Met deze optie kunt u een voorwaarde stellen waaraan moet worden voldaan voordat gegevens worden opgenomen in de GET-variabele. Als niet aan de voorwaarde wordt voldaan, verschijnt de melding **Niet voldaan aan bewerkingsvoorwaarde**, of de melding die u zelf hebt opgegeven met de optie ERROR.

OPMERKING
Als een door de gebruiker gedefinieerde functie resulteert in een logische waarde, kunt u de betreffende functie ook gebruiken als voorwaarde voor VALID. Zo kunt u op veel verschillende manieren gegevens controleren. Raadpleeg hoofdstuk 1, "Basisbegrippen", voor meer informatie over door de gebruiker gedefinieerde functies.

WHEN - U kunt een voorwaarde opgeven waaraan moet worden voldaan wanneer u de cursor in een GET-veld wilt plaatsen. Als de voorwaarde waar is (.T.), wordt de cursor in het veld geplaatst zodat u het kunt bewerken. Als de voorwaarde onwaar is (.F.), wordt het veld overgeslagen en wordt de cursor in het volgende veld geplaatst.

Veldopmaak-functies

De meeste veldopmaak-functies kunt u gebruiken met de opties PICTURE of FUNCTION of met de functie TRANSFORM.

Als u een veldopmaak-functie in een PICTURE-clausule opneemt, moet het @-symbool het eerste teken van de clausule zijn. Als u een veldopmaak-functie bij de optie FUNCTION gebruikt, hoeft u het @-teken niet op te geven. Als u zowel een veldopmaak-functie als een template gebruikt in een PICTURE-clausule, moeten deze met een spatie van elkaar worden gescheiden.

dBASE IV kent de volgende veldopmaak-functies:

Tabel 2-2 De veldopmaak-functies van dBASE IV

Functie	Omschrijving
!	Alle tekens toegestaan; kleine letters worden omgezet in hoofdletters.
^	Getallen worden weergegeven in wetenschappelijke notatie.
$	Gegevens worden weergegeven in financiële notatie met valutateken.
(	Negatieve getallen worden tussen haakjes geplaatst.
A	Alleen alfabetische tekens.
B	Numerieke waarde wordt links uitgelijnd (alleen in @...SAY).
C	Achter positieve getallen wordt CR (credit) weergegeven.
D	De actieve SET DATE-opmaak voor datums.
E	Europese datumindeling.
I	Tekst wordt gecentreerd (alleen in @...SAY).
J	Tekst wordt rechts uitgelijnd (alleen in @...SAY).
L	Voorloopnullen worden weergegeven.

Tabel 2-2 De veldopmaak-functies van dBASE IV

Functie	Omschrijving
M	Een keuzelijst voor GET-variabelen wordt gegeven.
R	Vaste tekens worden in een template weergegeven, maar niet in het veld ingevoerd.
S<n>	De breedte van het weergegeven veld wordt tot <n> tekens beperkt; de tekens worden horizontaal in het veld verschoven in <n> kolommen. <n> moet een positief geheel getal en een constante zijn.
T	De voorloopspaties en volgspaties in een veld worden verwijderd.
X	Achter negatieve getallen wordt DB (debet) weergegeven.
Z	De numerieke waarde nul wordt weergegeven als een lege tekenreeks.

Sommige functies kunnen alleen voor bepaalde gegevenstypen worden gebruikt:

- De functies ^ , $, (, C, L, X en Z kunnen alleen voor numerieke gegevens worden gebruikt (type N of type Z). Voorts kunnen (, C en X alleen worden gebruikt om gegevens weer te geven (dat wil zeggen, met de SAY-clausule). De functies $ en L kunnen worden gebruikt met SAY of GET.
- D en E kunnen alleen worden gebruikt voor datumgegevens.
- A, M, R en S<n> hebben alleen betrekking op tekengegevens.

Door de functies uit de tabel met elkaar te combineren kunt u nieuwe veldopmaak-functies definiëren. Met de functie XC bijvoorbeeld wordt DB weergegeven achter negatieve getallen en CR achter positieve getallen. Sommige functies, zoals D en E, kunnen niet worden gecombineerd.

Met de functie $ wordt de uitdrukking weergegeven met een valutateken voor of achter het bedrag. U kunt deze functie alleen gebruiken in GET-commando's als SET CURRENCY op LEFT is ingesteld. U kunt het valutateken, de plaats van het valutateken, de scheidingstekens en het decimaalteken wijzigen met de commando's SET CURRENCY, SET SEPARATOR en SET POINT.

Met S<n> (zonder spatie tussen S en de variabele) kunt u tekenvariabelen van GET weergeven en bewerken. Het aantal kolommen dat u opgeeft met de constante <n> (een geheel getal) moet kleiner zijn dan de werkelijke lengte van het tekenveld of de geheugenvariabele. Plaats geen spaties tussen de S en het gehele getal en gebruik geen punthaken (<>).

De tekenreeks verschuift binnen de opgegeven breedte, zodat u de gehele tekenreeks kunt bekijken en bewerken. Met ← , →, **Home** en **End** maakt u de verborgen tekens zichtbaar. Bij het invoeren van gegevens verschuift de tekenreeks automatisch.

Met de functie M kunt u een keuzelijst gebruiken voor GET-variabelen. De functie heeft de volgende indeling:

FUNCTION "M <keuzelijst>"

De opties in de keuzelijst kunnen constante tekenreeksen of constante getallen zijn en moeten van elkaar worden gescheiden door komma's. Omdat de komma een nieuwe keuzemogelijkheid in de lijst aangeeft, mag u geen komma's gebruiken in de opties in de keuzelijst. Twee komma's achter elkaar in een lijst resulteren in een lege waarde.

OPMERKING
Als in de PICTURE-clausule van de GET-opdracht de veldopmaak-functies M en < beide worden opgegeven, heeft de meerkeuzefunctie prioriteit boven de hoofdletterfunctie. De tekst van de meerkeuze-opties wordt niet in hoofdletters omgezet wanneer deze op het scherm wordt weergegeven of in een variabele wordt opgeslagen.

Met GET wordt de eerste optie van de lijst weergegeven. Als de waarde van de GET-variabele niet gelijk is aan een van de waarden in de lijst, krijgt de variabele de eerste waarde uit de lijst wanneer u een READ-commando geeft. Druk op de **spatiebalk** om de resterende opties te bekijken, totdat u de gewenste waarde hebt gevonden. Druk op ↵ om een optie uit de lijst te selecteren en naar het volgende veld te gaan.

OPMERKING
Wanneer de eerste GET-variabele in het veld wordt weergegeven, gaan alle gegevens in een invoerveld verloren.

U kunt ook een optie uit de lijst selecteren door de eerste letter te typen. Als niet alle opties in de lijst een unieke beginletter hebben, wordt de eerstvolgende optie met de opgegeven beginletter geselecteerd.

Met de functies B, I en J worden korte velden binnen een template ingekort en uitgelijnd. Geeft u geen uitlijnoptie op, dan worden tekenreeksen links uitgelijnd en getallen rechts.

Templates

U maakt een template door voor ieder teken dat moet worden weergegeven of ingevoerd één symbool op te geven.

Als u de functie R gebruikt met een template dat behalve de symbolen van de template ook andere tekens bevat, worden de betreffende tekens wel weergegeven maar niet opgeslagen als onderdeel van de GET-variabele. Als u R niet gebruikt, worden de betreffende tekens weergegeven en wel opgeslagen in de GET-variabele. R kan alleen worden gebruikt voor tekenvariabelen. Bij numerieke variabelen worden alle tekens die geen symbool van de template zijn, altijd weergegeven maar niet opgeslagen als onderdeel van het getal. Voor datumvariabelen en logische variabelen kunt u alleen de template-symbolen gebruiken.

In dBASE IV zijn de volgende template-symbolen beschikbaar:

Tabel 2-3 Template-symbolen in dBASE IV

Symbool	Toegestaan in combinatie met
9	Voor tekengegevens; alleen cijfers (0-9). Voor numerieke gegevens; cijfers en + en -
#	Alleen cijfers, spaties, een punt (.), + en -
A	Alleen letters
N	Zowel letters als cijfers, ook (_); geen spaties en leestekens
Y	Alleen de logische waarden Y, y, N en n; y en n worden omgezet in hoofdletters
L	T, F, Y, N
X	Alle tekens
!	Alle tekens; kleine letters worden omgezet in hoofdletters
overige	Gebruik de R-functie als u andere symbolen in een template wilt gebruiken. Met R worden de constanten weergegeven in de template maar niet in het veld.

Er zijn nog vier symbolen die in numerieke templates kunnen worden gebruikt. Deze staan in tabel 2-4.

Tabel 2-4 Symbolen voor numerieke templates

Symbool	Omschrijving
.	Scheidt de gehele getallen van de decimalen met een punt (.) (of met een ander decimaalteken indien opgegeven bij SET POINT)
,	Scheidt de duizenden met een komma (of met een ander teken indien opgegeven bij SET SEPARATOR)
*	Vervangt de voorloopnullen door asterisken
$	Vervangt de voorloopnullen door de tekenreeks die bij SET CURRENCY is opgegeven

De symbolen <, #, 9, A, N en X kunnen worden gebruikt met SAY-en GET-commando's. Met de symbolen 9, #, A, N en X voorkomt u dat ongedefinieerde tekens worden ingevoerd, maar niet dat deze tekens worden weergegeven.

Als u een template gebruikt om een decimaal getal op te vragen, moet u het decimaalteken in de template opnemen. In de template moet ook ruimte worden opengelaten voor ten minste één cijfer links van het decimaalteken en voor het minteken.

Richtlijnen voor het programmeren

Als u een indelingsbestand (.fmt) van meerdere pagina's wilt gebruiken waarin de @...SAY...GETs doorlopen van pagina 2 tot en met pagina 32, neemt u een READ-commando op in het bestand op de plaats waar u een nieuwe pagina wilt beginnen. Met **PgDn** en **PgUp** kunt u van pagina naar pagina gaan. Indelingsbestanden met meerdere pagina's functioneren alleen als het .fmt-bestand is geopend met SET FORMAT.

Tip

Voor het ontwerpen van een aangepast formulier maakt u met CREATE/MODIFY SCREEN een indelingsbestand (.fmt) dat bestaat uit @-commando's.

U activeert het indelingsbestand met SET FORMAT TO <.fmt-bestandsnaam>. U kunt nieuwe records toevoegen of bestaande records van een .fmt-bestand bewerken met de bewerkcommando's, zoals EDIT en APPEND.

Voorbeelden

De informatie in het eerste record van de database Afnemers geeft u als volgt weer:

```
. USE Afnemers
. <CLEAR
. @ 5,0 SAY TRIM(Voornaam) + " " + Achternaam
```

Nadat u een commando hebt ingevoerd, wordt de resulterende uitdrukking op het scherm weergegeven. Om de weergave duidelijk leesbaar te maken, neemt u in de uitdrukking spaties tussen aanhalingstekens op, zoals in het voorgaande voorbeeld te zien is. Het resultaat wordt als volgt weergegeven op het scherm:

```
Fred Wijsman
```

Met de PICTURE functie @M maakt u een keuzelijst waaruit gebruikers in een programmabestand een keuze kunnen maken. Selecteer bijvoorbeeld een dag van de week als volgt:

```
SET STATUS ON   && STATUS moet op ON staan voor @...GET MESSAGE.
Dag = "Iedere"
@ 12,20 SAY "Selecteer de dag van de week " GET Dag;
            PICTURE "@M Ma,Di,Wo,Do,Vr,Za,Zo";
            MESSAGE "Spatiebalk: waarden bekijken  RETURN: selecteren"
READ
```

In dit voorbeeld wordt eerst "Iedere" in het invoerveld weergegeven. Zodra de cursor echter in het veld komt, wordt de weergave gewijzigd in de eerste optie van de lijst, "Ma". Hierna verschijnt "Iedere" niet meer omdat het niet in de lijst voorkomt.

Als u de VALID-clausule in combinatie met een door de gebruiker gedefinieerde functie opgeeft, waarborgt u de betrouwbaarheid van invoer. In het volgende voorbeeld voert u in een variabele met de naam Mtoeslag een bedrag in. Mtoeslag moet gelijk zijn aan 1, 2 of 3 voor respectievelijk een werkdag, zaterdag of zondag.

```
Mtoeslag = 0
@ 14,20 SAY "De toeslag is " GET Mtoeslag PICTURE "9";
    VALID Rcheck();
    ERROR "Werkdag = 1, Zaterdag = 2, Zondag = 3"
READ
RETURN

FUNCTION Rcheck
DO CASE
    CASE Dag <> "Z" .AND. Mtoeslag = 1
    RETURN .T.
    CASE Dag = "Za" .AND. Mtoeslag = 2
    RETURN .T.
    CASE Dag = "Zo" .AND. Mtoeslag = 3
    RETURN .T.
ENDCASE
RETURN .F.
```

In bovenstaand voorbeeld roept de VALID-voorwaarde de door de gebruiker gedefinieerde functie Rcheck() op. Als de invoer aan de voorwaarden voldoet, resulteert Rcheck() in waar (.T.). Als Rcheck() onwaar (.F.) is, wordt de foutmelding **Werkdag = 1, Zaterdag = 2, Zondag = 3** weergegeven en mag de gebruiker het veld niet verlaten voordat de fout is hersteld.

Zie ook

?/??, ACTIVATE WINDOW, APPEND, CHANGE, COL(), CREATE/MODIFY SCREEN, EDIT, INSERT, MODIFY COMMAND, PCOL(), PROW(), READ, ROW(), SET COLOR, SET CONFIRM, SET CURRENCY, SET DELIMITERS, SET DEVICE, SET FIELDS, SET FORMAT, SET INTENSITY, SET POINT, SET SEPARATOR, SET WINDOW OF MEMO, TRANSFORM()

@...CLEAR

Met @...CLEAR wordt een gedeelte van het beeldscherm of van het actieve venster gewist.

Syntaxis

@ <rij1>, <kolom1> CLEAR [TO <rij2>, <kolom2>]

Gebruik

<rij1>,<kolom1> zijn de coördinaten van de linker bovenhoek van het gebied dat u wilt wissen en <rij2>,<kolom2> zijn de coördinaten van de rechter benedenhoek.

Met dit commando wordt een deel van het scherm gewist. Het te wissen gedeelte loopt van <rij1>, <kolom1> tot en met <rij2>, <kolom2>. Als u CLEAR TO <rij2>, <kolom2> niet opneemt in het commando, wordt de regel die begint op <rij1>, <kolom1> gewist tot aan het einde van de regel. Als u TO <rij2>, <kolom2> weglaat maar wel het sleutelwoord CLEAR gebruikt, wordt het beeldscherm of het actieve venster gewist van <rij1>, <kolom1> tot aan de rechter benedenhoek.

Voorbeeld

Om het deel van het scherm vanaf de coördinaten 2,9 tot en met 14,39 te wissen, gaat u als volgt te werk:

```
@ 2,9 CLEAR TO 14,39
```

Zie ook

@...FILL, CLEAR

Met @...FILL kunt u in een bepaald gedeelte van het scherm of in het actieve venster de kleuren wijzigen.

Syntaxis

@ <rij1>,<kolom1> FILL TO <rij2>,<kolom2>

[COLOR <kleurattribuut>]

Gebruik

Met dit commando wordt de kleur van de tekst in het gedefinieerde gebied gewijzigd. <rij1>,<kolom1> zijn de coördinaten van de linker bovenhoek van het gebied en <rij2>,<kolom2> zijn de coördinaten van de rechter benedenhoek.

Bij <kleurattribuut> geeft u de kleurcodes op voor het gebied. Dit zijn dezelfde codes die worden gebruikt in SET COLOR.

U kunt alleen de standaard voor- en achtergrondkleur voor het gebied wijzigen. Dit commando beïnvloedt alleen de gegevens die al op het scherm worden weergegeven. Als u hierna met een ander commando gegevens naar dit gebied schrijft, worden de standaardkleuren weer gebruikt in plaats van de kleuren die u hebt ingesteld met @...FILL.

Als u de optie COLOR weglaat, wordt met @...FILL het opgegeven gebied van het scherm gewist, net als met het commando @...CLEAR.

Als u coördinaten opgeeft die buiten het beeldscherm vallen, verschijnt de foutmelding **Coördinaten liggen buiten het scherm**.

Voorbeeld

U wilt de kleuren van het gebied met de coördinaten 3,10 tot en met 20,70 wijzigen in rood op een zwarte achtergrond. Vul het scherm eerst echter met tekst met het commando LIST MEMORY:

```
. LIST MEMORY
. @ 3,10 FILL TO 20,70 COLOR R/N
```

Zie ook

@...CLEAR, SET COLOR

@...TO

Met @...TO wordt een kader op het scherm of in het actieve venster getekend dat bestaat uit een enkele lijn, dubbele lijnen of speciale door u opgegeven tekens.

Syntaxis

@ <rij1>,<kolom1> TO <rij2>,<kolom2>

[DOUBLE/PANEL/>kaderdefinitie>]

[COLOR <kleurattribuut>]

Standaardinstellingen

De standaardinstelling voor het kader is een enkele lijn, tenzij u deze instelling hebt gewijzigd met het commando SET BORDER.

De standaardkleur is NORMAL. Deze kleur kunt u ook wijzigen door een waarde voor het sleutelwoord NORMAL in het commando SET COLOR op te geven.

Gebruik

<rij1>,<kolom1> zijn de coördinaten van de linker bovenhoek van het kader en <rij2>,<kolom2> zijn de coördinaten van de rechter benedenhoek.

Als de rijcoördinaten gelijk zijn, wordt een horizontale lijn getekend. Als de kolomcoördinaten gelijk zijn, wordt een verticale lijn getekend.

Een kader dat is gedefinieerd met de opties van het commando @...TO heeft een hogere prioriteit dan de standaardinstelling SET BORDER.

OPMERKING
Met @...TO wordt geen gegevensstroom geproduceerd, maar de positie van de daarop volgende uitvoer kan door dit commando wel worden beïnvloed. Als u bijvoorbeeld de positie van de cursor op het scherm met het commando @...TO wijzigt, is dit van invloed op de positie waar een later ?/??-commando wordt weergegeven.

Opties

Met DOUBLE wordt het kader met dubbele lijnen getekend in plaats van met enkele lijnen.

Met PANEL wordt een gemarkeerd kader weergegeven in omgekeerde weergave.

<kaderdefinitie> is een lijst met tekenreeksen (of getallen) waarmee u een kader kunt definiëren. De tekenreeksen moeten van elkaar worden gescheiden door komma's en moeten in de volgende volgorde staan:

b,o,l,r,lb,rb,lo,ro

De letters geven de volgende attributen aan:

b - bovenlb - linker bovenhoek

o - onder rb - rechter bovenhoek

l - links lo - linker benedenhoek

r - rechts bo - rechter benedenhoek

Als u alleen het eerste attribuut (b) opgeeft, krijgen de overige attributen standaard dezelfde waarde.

U kunt een attribuut weglaten door dit te vervangen door een komma als het attribuut aan het begin van een lijst staat, of door het attribuut weg te laten als het aan het einde van een lijst staat. Als u een attribuut weglaat, wordt de oorspronkelijke instelling niet gewijzigd.

Gebruikt u getallen in plaats van tekenreeksen, dan moet u de decimale waarde van het teken opgeven uit de IBM Extended Character Set. Merk op dat de waarden van de ASCII-code (0 tot en met 127 decimaal) onderdeel vormen van deze tekenset (0 tot en met 255 decimaal). U mag ook getallen invoeren als argument van de functie CHR(). De getallen mogen niet tussen scheidingstekens worden geplaatst.

COLOR - Bij <kleurattribuut> geeft u kleurcodes op voor de achtergrond, de voorgrond of voor beide. Dit zijn dezelfde codes die worden gebruikt in SET COLOR. Als u de optie SET PANEL hebt opgegeven, wordt het venster alleen in de voorgrondkleur getekend. Als u geen kleurcodes opgeeft, worden de NORMAL-kleuren van het commando SET COLOR gebruikt.

Richtlijnen voor het programmeren

U kunt het commando @...TO opgeven vanaf de commandostip of in een commando- of indelingsbestand.

Om te voorkomen dat het kader wordt overschreven door een veld dat breder is dan het venster, beperkt u de afmetingen van het invoergebied op het scherm met de functie S<n> van het @-commando. Met de functie S<n> kunt u horizontaal verschuiven binnen het invoergebied.

Om een kader af te drukken, gebruikt u het DEFINE BOX commando.

Voorbeeld

Als u een kader met dubbele lijnen wilt tekenen tussen de coördinaten 1,10 en 15,40 met de kleurattributen zwart op cyaan, gaat u als volgt te werk:

```
. @ 1,10 TO 15,40 DOUBLE COLOR N/BG
```

Zie ook

@, @...CLEAR, @...FILL, CHR(), DEFINE BOX, SET BORDER, SET COLOR

ACCEPT

ACCEPT wordt hoofdzakelijk gebruikt in commandobestanden. Met ACCEPT wordt de gebruiker gevraagd gegevens vanaf het toetsenbord in te voeren. Met dit commando wordt een tekengeheugenvariabele gemaakt waarin de toetsenbordinvoer wordt opgeslagen. Beëindig de invoer van gegevens met ↵.

Syntaxis

ACCEPT [<aanwijzing>] TO <geheugenvariabele>

Gebruik

<aanwijzing> kan een tekengeheugenvariabele zijn, een tekenreeks of elke geldige tekenuitdrukking. Als <aanwijzing> een tekenreeks is, moeten de elementen van de reeks van elkaar worden gescheiden door enkele aanhalingstekens (''), dubbele aanhalingstekens ("") of vierkante haken ([ ]).

Voor de invoer via het toetsenbord hoeven geen scheidingstekens te worden gebruikt. ACCEPT beschouwt alle invoer van de gebruiker als tekengegevens.

Wanneer u op ↵ drukt als antwoord op het commando ACCEPT, wordt de inhoud van de geheugenvariabele nul (zonder inhoud of ASCII 0).

Met ACCEPT kunnen maximaal 254 tekens in een variabele worden ingevoerd.

Richtlijnen voor het programmeren

Tenzij SET ESCAPE op OFF is ingesteld, wordt het programma beëindigd als u op **Esc** drukt als antwoord op een ACCEPT-commando.

Voorbeeld

Als u de gebruiker wilt vragen zijn sofinummer in te voeren en als u deze gegevens wilt opslaan in de geheugenvariabele Sofi, doet u het volgende:

```
. ACCEPT "Voer uw sofinummer in:" TO Sofi
Voer uw sofinummer in:
```

Zie ook

@, INPUT, READ, SET ESCAPE, STORE, WAIT

ACTIVATE MENU

Met dit commando wordt een bestaand balkmenu geactiveerd en op het scherm weergegeven.

Syntaxis

ACTIVATE MENU <menunaam> [PAD <stripnaam>]

Gebruik

Met dit commando activeert u een eerder gedefinieerd menu, dat op het scherm over eventueel aanwezige gegevens heen wordt weergegeven. Door in het commando ACTIVATE MENU de optie PAD op te geven, verschijnt de gemarkeerde balk in de strip. Doet u dit niet, dan wordt de eerste gedefinieerde strip gemarkeerd.

Het menu dat het laatst is geactiveerd is het enige menu dat actief is. Gebruik de toetsen → en ← om van de ene menustrip naar de andere te gaan. De strips worden doorlopen in de volgorde waarin deze zijn gedefinieerd. U kunt een actief menu inactief maken door een ander menu te activeren met **Esc** of door het commando DEACTIVATE MENU te geven.

Wanneer een menu door een ander menu is geactiveerd, wordt het eerste menu onderbroken totdat het tweede menu inactief wordt gemaakt.

Voorbeeld

. ACTIVATE MENU Hoofdmenu PAD Bekijken

Het menu met de naam Hoofdmenu wordt geactiveerd en de cursor wordt in de eerste strip met de naam Bekijken geplaatst. Bij het voorbeeld van DEFINE PAD in dit hoofdstuk kunt u lezen hoe u Hoofdmenu en Bekijken instelt.

Zie ook

DEACTIVATE MENU, DEFINE MENU, DEFINE PAD, MENU(), ON PAD, PAD(), SHOW MENU

ACTIVATE POPUP

Met dit commando activeert u een eerder gedefinieerd pop-up menu.

Syntaxis

ACTIVATE POPUP <pop-up naam>

Gebruik

U kunt slechts één pop-up menu tegelijkertijd activeren. Als een pop-up menu actief is, wordt dit inactief gemaakt wanneer u het commando ACTIVATE POPUP nogmaals geeft, op **Esc** drukt of het commando DEACTIVATE POPUP gebruikt.

Wanneer een pop-up menu door een ander pop-up menu wordt geactiveerd, wordt het eerste pop-up menu onderbroken totdat het tweede menu inactief wordt gemaakt.

Voorbeeld

Met dit commando activeert u een eerder gedefinieerd pop-up menu dat enkele opties bevat om het programma te verlaten:

```
. ACTIVATE POPUP Stop_pop
```

Zie ook

BAR(), DEACTIVATE POPUP, DEFINE POPUP, POPUP(), PROMPT(), SHOW POPUP

ACTIVATE SCREEN

Zodra u het commando ACTIVATE SCREEN hebt gegeven, kunt u weer over het gehele scherm beschikken en niet alleen over het (kleinere) actieve venster. Het venster dat het laatst actief was blijft op het scherm, maar kan worden overschreven, verschoven of gewist.

Syntaxis

ACTIVATE SCREEN

Gebruik

ACTIVATE SCREEN zendt de uitvoer naar het volledige beeldscherm in plaats van naar het actieve venster. Het actieve venster verdwijnt niet en u hoeft ook geen venster te definiëren dat samenvalt met de coördinaten van het volledige beeldscherm.

U kunt ACTIVATE SCREEN gebruiken wanneer u schermgroot met code wilt werken en toch een venster op het scherm wilt laten staan. Ook pop-up menu's blijven op de juiste relatieve positie staan.

ACTIVATE SCREEN wordt ook gebruikt om tekst buiten een venster te plaatsen, bijvoorbeeld wanneer u een extra melding wilt weergeven voor de gebruiker. In dat geval gebruikt u ACTIVATE WINDOW nadat u ACTIVATE SCREEN hebt gebruikt om de uitvoer naar het beeldscherm tijdelijk te wijzigen.

Hoewel het actieve venster op de voorgrond blijft, wordt de inhoud niet bijgewerkt. De eventuele uitvoer van het venster wordt naar het volledige scherm gezonden, dus ook naar het gedeelte dat door het venster in beslag wordt genomen. Wanneer de uitvoer naar het volledige scherm wordt gezonden, kan het zijn dat de tekst over het venster wordt geschreven of dat het venster van het scherm wordt geschoven. Met het commando CLEAR kunt u het scherm volledig wissen.

Het venster blijft op het scherm totdat dit inactief wordt gemaakt. Wanneer u het venster daarna opnieuw activeert, bedekt het de op die positie aanwezige tekst.

Zie ook

@...FILL, ACTIVATE WINDOW, CLEAR, DEACTIVATE WINDOW, DEFINE WINDOW

ACTIVATE WINDOW

Met het commando ACTIVATE WINDOW activeert u een gedefinieerd venster uit het geheugen, waarna dit venster op het scherm wordt weergegeven. De beeldschermuitvoer wordt naar dat venster gezonden.

Syntaxis

ACTIVATE WINDOW <vensternamenlijst>/ALL

Gebruik

Om ACTIVATE WINDOW te kunnen gebruiken, moet ten minste één gedefinieerd venster in het geheugen zijn opgeslagen. Meerdere vensters kunnen op het scherm staan, maar slechts één venster kan actief zijn. Als u een lijst met vensternamen opgeeft, is het laatste venster in de lijst actief.

Wanneer u de optie ALL gebruikt, worden alle gedefinieerde vensters die op dat moment in het geheugen aanwezig zijn, weergegeven in de volgorde waarin deze zijn gedefinieerd. De grootte van de vensters komt overeen met de opmaak die u bij het definiëren van de vensters hebt gekozen. Als u bij het definiëren van het venster geen tekenreeks voor de rand opgeeft, worden de instellingen van SET BORDER gehanteerd.

Voorbeeld

Met het volgende commando wordt een reeds gedefinieerd venster met de naam W1 geactiveerd:

```
. ACTIVATE WINDOW W1
```

Zie ook

CLEAR WINDOW, DEACTIVATE WINDOW, DEFINE WINDOW, FUNCTION, MOVE WINDOW, RESTORE WINDOW, SAVE WINDOW, SET BORDER

APPEND

Met APPEND kunt u nieuwe records toevoegen achter aan het actieve database-bestand.

Syntaxis

APPEND [BLANK]

Gebruik

Met het commando APPEND activeert u de schermgrote gegevensinvoermodus. In een *schermformulier* wordt één nieuw record tegelijkertijd afgebeeld. Als geen indeling is opgegeven, wordt een standaardformulier weergegeven.

U beëindigt het proces door op ↵ of **Esc** te drukken zodra een nieuw record wordt afgebeeld. Als u naar een nieuw record gaat en op **Ctrl-End** drukt, wordt een leeg record aan het bestand toegevoegd.

Met **PgUp** verplaatst u de cursor naar voorgaande records die u vervolgens kunt bewerken. Als er een index is geopend, staan de records in indexvolgorde. Als dat niet het geval is, staan ze in recordvolgorde. Met **PgDn** verplaatst u de cursor vooruit door de records. Na het laatste record keert u dan terug naar de APPEND-modus.

Met **F10** kunt u schakelen tussen het APPEND-scherm en het bijbehorende balkmenu.

U kunt een memoveld invoeren door de cursor op het woord *memo* te plaatsen en op **Ctrl-Home** te drukken. Als u de tekstverwerker van dBASE IV gebruikt om in een memoveld te werken, heeft u de beschikking over dezelfde stuurtoetsen als bij MODIFY COMMAND.

Met APPEND kunt u slechts aan één database-bestand records toevoegen. Met behulp van een indelingsbestand kunt u met @...GET velden opvragen uit meerdere gekoppelde bestanden. Wanneer u APPEND in combinatie met een indelingsbestand gebruikt, worden geen records toegevoegd aan bestanden die niet zijn geselecteerd. U kunt echter het commando READ met een indelingsbestand gebruiken om gegevens toe te voegen aan records uit meerdere bestanden tegelijk. Om een record aan meerdere bestanden toe te voegen, kiest u eerst APPEND met de optie BLANK en geeft u vervolgens het commando READ.

Opties

Met de optie BLANK wordt een leeg record achter aan het database-bestand toegevoegd, maar wordt de gegevensinvoermodus niet geactiveerd. Het lege record wordt het actieve record.

Tips

Wanneer records worden toegevoegd, worden alle actieve indexen (inclusief .mdx-labels) bijgewerkt.

Voorbeelden

Om de gegevensinvoermodus te activeren en records toe te voegen aan het database-bestand Afnemers, voert u het volgende in:

```
. USE Afnemers
. APPEND
```

Om een leeg record toe te voegen aan het database-bestand Afnemers zonder de gegevensinvoermodus te activeren, voert u het volgende in:

```
. APPEND BLANK
```

Zie ook

@...GET, BROWSE, EDIT, SET AUTOSAVE, SET CARRY, SET FORMAT, SET WINDOW OF MEMO

APPEND FROM

Met APPEND FROM worden records uit een bestaand bestand naar het actieve database-bestand gekopieerd. Het bestand van waaruit wordt gekopieerd hoeft geen dBASE IV-bestand te zijn.

Syntaxis

APPEND FROM <bestandsnaam>/?
[[TYPE] <bestandstype>] [FOR <voorwaarde>]

Standaardinstellingen

Als het bestand niet in de standaardbestandsindex is opgeslagen, moet de bestandsnaam het station bevatten waar het bestand is opgeslagen, tenzij een pad naar het betreffende station is gedefinieerd.

Als u achter de bestandsnaam geen extensie opgeeft, gebruikt dBASE IV de extensie .dbf.

Als u geen extensie opgeeft achter de bestandsnaam, maar SDF of DELIMITED opgeeft, wordt automatisch de extensie .txt aan de bestandsnaam toegevoegd. dBASE IV gaat er tevens van uit dat andere bestandstypen de standaardextensie hebben die is gedefinieerd door het programma waartoe de betreffende bestanden behoren.

Gebruik

Als het bestand van waaruit (FROM) wordt gekopieerd een database-bestand van dBASE IV is (.dbf):

- worden records die van een wismarkering zijn voorzien *wel* toegevoegd en in de actieve database niet van wismarkering voorzien als SET DELETION op OFF is ingesteld. Als SET DELETION op ON is ingesteld, worden alleen records zonder wismarkering toegevoegd;
- worden alleen veldnamen die in beide bestanden worden aangetroffen, toegevoegd. De veldnamen hoeven niet in dezelfde volgorde te staan;
- geeft u *geen* bestandstype op.

Als een veld in het bestand van waaruit wordt gekopieerd groter is dan hetzelfde veld in de actieve database, gaan alle tekens die niet in het veld passen, verloren. Er worden dan geen numerieke gegevens toegevoegd. Als het bestand waaraan wordt toegevoegd een geopend indexbestand heeft, wordt de index automatisch bijgewerkt.

Met een FOR-voorwaarde kunt u alleen die velden testen die zich bevinden in beide bestanden die dezelfde naam hebben. Met een FOR-clausule wordt een record in het database-bestand van waaruit wordt gekopieerd overigens alleen aan het actieve database-bestand toegevoegd als het resulterende record aan de FOR-criteria voldoet. Aangezien de FOR-voorwaarde op het nieuwe record wordt toegepast in plaats van op het record in het database-bestand van waaruit wordt gekopieerd, is het mogelijk dat u onverwachte resultaten krijgt. Hierna volgen twee voorbeelden die dit illustreren.

Voorbeeld 1:

De waarde die de functie RECNO() in de opdracht APPEND FROM Nieuwdb FOR RECNO() = 100 oplevert, is het eerste beschikbare recordnummer in het actieve database-bestand en niet het recordnummer in het database-bestand van waaruit wordt gekopieerd. In dit voorbeeld wordt alleen een record toegevoegd als de actieve database exact 99 records bevat.

Voorbeeld 2:

Als u probeert met de opdracht APPEND FROM Test FOR DELETED() alleen de records met een wismarkering uit het database-bestand Test toe te voegen, controleert u in werkelijkheid alleen of het actieve bestand (het actieve, geopende bestand) wismarkeringen bevat. In dit voorbeeld worden alle records van Test toegevoegd, ongeacht hun wismarkering.

Opties

De opties voor <bestandstype> zijn:

- DBASEII - dBASE II database-bestand.
- DELIMITED - Delimited Format ASCII-bestand. Gegevens worden van links naar rechts teken voor teken toegevoegd. Elk record moet eindigen met een terugloop- en een regeldoorvoerteken. De velden moeten van elkaar worden gescheiden door komma's en tekengegevens moeten worden gescheiden door dubbele aanhalingstekens, tenzij u een ander scheidingsteken hebt opgegeven. Deze optie is gelijk aan DELIMITED WITH ".
- DELIMITED WITH BLANK werkt met bestanden waarin de velden worden gescheiden door één spatie. Er staan geen komma's tussen de velden en elk record eindigt met een terugloop- en een regeldoorvoerteken.
- DELIMITED WITH <scheidingsteken> werkt met delimited ASCII-bestanden waarin de velden niet worden gescheiden door komma's en waarbij tekenreeksen tussen dubbele aanhalingstekens staan, en elke regel wordt beëindigd met een wagenterugloop en een regeldoorvoerteken.
- DIF - VisiCalc bestandsindeling. De VisiCalc-rijen worden omgezet in records en de kolommen worden omgezet in velden.
- FW2 - Framework IITM database- of spreadsheet-frame.
- RPD - RapidFileTM-gegevensbestand.
- SDF - System Data Format ASCII-bestand. Dergelijke bestanden worden ook wel *fixed-length*-bestanden genoemd. Gegevens worden van links naar rechts teken voor teken toegevoegd. Elk record in het bronbestand heeft dezelfde lengte en eindigt met een terugloop- en een regeldoorvoerteken. Afzonderlijke velden worden niet voorzien van een scheidingsteken.
- SYLK - MultiPlan-spreadsheet in *oplopende* volgorde. De MultiPlan-rijen worden omgezet in records en de kolommen worden omgezet in velden. dBASE IV kan geen records overnemen uit een SYLK-bestand dat is opgeslagen in *afdalende* volgorde.

- WKS - Lotus 1-2-3 spreadsheet, versie 1A. De rijen van Lotus 1-2-3 worden omgezet in records en de kolommen worden omgezet in velden. Het bestand begint met de cel in de linker bovenhoek van het spreadsheet.

Blanco rijen in alle spreadsheets worden omgezet in lege records in het database-bestand.

Wanneer u APPEND FROM gebruikt om een van de genoemde spreadsheets te lezen, moeten de gegevens die u inleest met de structuur van het database-bestand overeenkomen. Dit betekent dat het spreadsheet in *oplopende volgorde* (in plaats van in *afdalende volgorde* moet zijn opgeslagen en dat de titels van de kolommen moeten zijn verwijderd voordat u probeert de gegevens in dBASE IV in te lezen. Met APPEND FROM worden wel de namen van rijen opgeslagen, als daar tenminste rekening mee is gehouden bij het ontwerpen van de database-structuur. Bestanden uit Lotus 1-2-3 mogen niet beginnen met blanco rijen of kolommen. Is dat wel het geval, dan moet u de gegevens eerst naar links of naar boven verplaatsen voordat u APPEND FROM kunt toepassen.

Tip

Als u APPEND gebruikt met een bestand waarvoor dBASE IV een bepaalde extensie verwacht terwijl dit bestand geen extensie heeft, typt u een punt achter de bestandsnaam.

Speciale gevallen

Met het commando IMPORT kunt u bestanden van PFS:FILE en van Lotus release 2.x converteren naar dBASE IV-bestanden. U kunt IMPORT ook gebruiken om bestanden van dBASE II, Framework II en RapidFile te converteren naar dBASE IV-bestanden.

In datumvelden kunnen met het commando APPEND FROM gegevens worden overgenomen uit tekenvelden als de gegevens de juiste datumindeling hebben. Omgekeerd kunnen in tekenvelden (als deze groot genoeg zijn en op de juiste manier zijn ingedeeld) gegevens worden overgenomen uit datumvelden. Uit tekstbestanden kunnen datums alleen worden overgenomen met APPEND FROM als de datums de vorm JJJJMMDD hebben en niet voorzien zijn van scheidingstekens (JJJJ is het jaar, MM is de maand en DD is de dag).

APPEND FROM

Voorbeeld

Om met het commando APPEND FROM gegevens uit een RapidFile-database met de naam Contact aan het gegevensbestand Afnemers toe te voegen, voert u het volgende in:

```
. USE Afnemers INDEX KIntnaam
. APPEND FROM Contact.rpd TYPE RPD
```

In dit voorbeeld wordt het bestand Klantnaam.ndx geopend om de betrouwbaarheid van de index te waarborgen. Als dat niet gebeurt, moet Klantnaam.ndx met REINDEX opnieuw worden geïndexeerd zodra het bestand weer wordt geopend.

Zie ook

COPY, DELETED(), IMPORT, SET DELETED

APPEND FROM ARRAY

Met APPEND FROM ARRAY worden records aan een database-bestand toegevoegd aan de hand van informatie in een reeks.

Syntaxis

APPEND FROM ARRAY <naam reeks> [FOR <voorwaarde>]

Gebruik

De inhoud van elke rij in de opgegeven reeks wordt een nieuw record in het actieve database-bestand.

Voor iedere rij in de reeks wordt de inhoud van de eerste kolom in het eerste veld geplaatst, de inhoud van de tweede kolom in het tweede veld, enzovoort. Dit proces wordt herhaald totdat er geen kolommen of geen velden meer zijn.

Als een reeks meer kolommen bevat dan de database velden heeft, worden de resterende kolommen van de reeks genegeerd. Als het database-bestand meer velden heeft dan de reeks kolommen heeft, blijven de resterende velden leeg.

Elk element van de reeks moet van hetzelfde gegevenstype zijn als het veld waarin het wordt geplaatst.

Als u de FOR-clausule gebruikt, wordt de voorwaarde verwerkt voordat elke rij van de reeks aan het database-bestand wordt toegevoegd. De <voorwaarde> in de FOR-clausule moet een veldnaam bevatten van een database-bestand. Een rij wordt alleen toegevoegd als de voorwaarde waar (.T.) is. Als de voorwaarde onwaar (.F.) is, wordt de rij overgeslagen en wordt de volgende rij verwerkt.

Als u een kolom met reekselementen op een bepaalde voorwaarde wilt controleren, geeft u de naam van het database-veld op die overeenkomt met het kolomnummer van de reeks en niet het kolomnummer zelf. In het gedeelte Voorbeelden wordt deze situatie nader toegelicht.

U moet de reeks eerst definiëren met het commando DECLARE en informatie opslaan in de elementen van de reeks met het commando STORE voordat u met APPEND gegevens uit de tabel kunt overnemen.

APPEND FROM ARRAY

Voorbeelden

Om gegevens uit een reeks aan het database-bestand Transact toe te voegen, moet u de reeks eerst definiëren:

```
. DECLARE Aanvulling[1,5]
. Aanvulling[1,1] = "M00001"
M00001
. Aanvulling[1,2] = "88-100"
88-100
. Aanvulling[1,3] = DATE()
01-03-87
. Aanvulling[1,4] = .F.
.F.
. Aanvulling[1,5] = 125.00
          125,00
. USE Transact
. APPEND FROM ARRAY Aanvulling
    1 record toegevoegd
```

In dit voorbeeld wordt gedemonstreerd hoe u een kolom met reekselementen op een bepaalde voorwaarde controleert:

```
. DECLARE Tijdelijk[5,11]
Tijdelijk[1,1] = "Henk"
Tijdelijk[2,1] = "Harm"
Tijdelijk[3,1] = "Hans"
Tijdelijk[4,1] = "    "
Tijdelijk[5,1] = "Huub"
USE Reizen                          && Eerste veld heet Voornaam
* Alleen records zonder spaties uit reeks toevoegen.
APPEND FROM ARRAY Tijdelijk FOR Voornaam = "H"
* 4 records toegevoegd
```

Zie ook

APPEND, COPY TO ARRAY, DECLARE, STORE

APPEND MEMO

Met APPEND MEMO importeert u een bestand in een benoemd memoveld van het actieve, geselecteerde record.

Syntaxis

APPEND MEMO <naam memoveld> FROM <bestandsnaam> [OVERWRITE]

Standaardinstellingen

Als u geen extensie bij de bestandsnaam opgeeft, wordt de standaardextensie .txt gebruikt.

Gebruik

Met APPEND MEMO kunt u ieder bestand naar een memoveld kopiëren. Het gehele bestand wordt gekopieerd en toegevoegd aan de bestaande inhoud van het memoveld.

Met de optie OVERWRITE kunt u de bestaande inhoud wissen voordat het bestand naar het memoveld wordt gekopieerd.

Als het benoemde veld niet wordt gevonden, ziet u de melding **Veld niet gevonden**.

Als het benoemde veld geen memoveld is, ziet u de melding **Veld moet memoveld zijn**.

Als de opgegeven bestandsnaam niet wordt gevonden, ziet u de melding **Bestand bestaat niet**.

Voorbeelden

Om een tekstbestand met de naam Nieuwtxt.txt toe te voegen aan het eerste record van het database-bestand Afnemers, moet u eerst het tekstbestand maken met het commando MODIFY FILE:

```
. MODIFY FILE Nieuwtxt.txt
Nieuwe tekst begint hier...
```

APPEND MEMO

Vervolgens slaat u het bestand op met **Ctrl-End**. Daarna typt u het volgende achter de commandostip:

```
. USE Afnemers
. ? Overzicht
85-200 02-08-85
    C-300-400 BOEKENKAST          535,00 1
. APPEND MEMO Overzicht FROM Nieuwtxt
. ? Overzicht
85-200 02-08-85
    C-300-400 BOEKENKAST          535,00 1
Nieuwe tekst begint hier...
```

Zie ook

COPY MEMO, MODIFY COMMAND/FILE

BEGIN/END TRANSACTION

De commando's BEGIN TRANSACTION en END TRANSACTION vormen respectievelijk het begin en het einde van een transactie; een reeks commando's die als een aparte eenheid worden beschouwd. Het verwerken van meerdere commando's in de vorm van transacties garandeert dat of alle of geen van de commando's die effect hebben op inhoud van een database-bestand, worden uitgevoerd.

Syntaxis

BEGIN TRANSACTION [<padnaam>]
<transactiecommando's>
END TRANSACTION

Gebruik

Nadat u het commando BEGIN TRANSACTION hebt gegeven, wordt een transactielog-bestand gemaakt en begint het vastleggen van de transactiegegevens. De transactie wordt niet eerder beëindigd dan wanneer u een commando END TRANSACTION, ROLLBACK, CANCEL of QUIT geeft. Gedurende de transactie worden records die zijn toegevoegd of gewijzigd opgeslagen, evenals mogelijke nieuwe bestanden die zijn gemaakt.

Wanneer u het commando SUSPEND geeft in een programma dat draait als onderdeel van een transactie, wordt niet de transactie onderbroken, maar het programma dat binnen de transactie draait. Nadat u het commando RESUME hebt gegeven, wordt een vervolg gemaakt met de uitvoering van de transactie.

In een programma heeft het commando ROLLBACK tot gevolg dat de besturing wordt overgedragen aan het commando dat direct volgt op END TRANSACTION. Commando's die tussen ROLLBACK en END TRANSACTION worden gegeven, worden genegeerd.

Het transactielog-bestand krijgt de naam Translog.log op een zelfstandig systeem en de naam <werkstationnaam>.log in een lokaal netwerk. <werkstationnaam> is de netwerknaam van de computer die het commando BEGIN TRANSACTION geeft. Als er geen <werkstationsnaam> beschikbaar is, wordt de aanmeldingsnaam gebruikt die bij PROTECT is opgegeven.

Volgens de standaardinstelling wordt het transactielog-bestand gemaakt in de actieve bestandsindex van het actieve station. Met de optie <padnaam> kunt u desgewenst een andere bestandsindex opgeven. Als de actieve bestandsindex zich op de file server van een netwerk bevindt, kunt u de prestatie in dat netwerk verbeteren door voor het transactielog-bestand een bestandsindex op te geven op het actieve werkstation.

Wanneer u BEGIN TRANSACTION en END TRANSACTION in een programma gebruikt, kunnen beide commando's niet onafhankelijk van elkaar worden gebruikt. Beide moeten in dezelfde procedure zijn opgenomen en *samen* deel uitmaken van één en hetzelfde geneste opdrachtblok van een procedure. Om die reden kan END TRANSACTION niet conditioneel worden gebruikt.

Anders dan bij andere gestructureerde commando's geldt voor transacties dat ze niet kunnen worden genest. U moet een transactie afronden voordat u een volgende kunt starten.

BEGIN/END TRANSACTION

Bij verwerking van transacties worden de functies COMPLETED() en ROLLBACK() gebruikt om de uitkomst van een transactie te testen.

Met COMPLETED() kunt u testen of een transactie met succes is afgerond. De functie levert waar (.T.) op, tenzij een transactie nog in uitvoering is. Met CANCEL geeft u opdracht tot uitvoering van de functie ROLLBACK, waardoor de database wordt teruggebracht in de oorspronkelijk staat van voor de transactie.

Met ROLLBACK() controleert u of ROLLBACK succesvol is geweest. De functie levert volgens de standaardinstelling waar (.T.) op. Indien zich een fout voordoet tijdens ROLLBACK, levert de functie onwaar (.F.) op, totdat een volgend ROLLBACK-commando wel succesvol is of totdat u een RESET-commando geeft.

Als het onmogelijk blijkt met het commando ROLLBACK een aantal database-bestanden als groep te herstellen, moet u de bestanden individueel proberen te herstellen.

Commando's die niet zijn toegestaan in transacties

De volgende commando's zijn niet toegestaan tijdens transacties:

CLEAR ALL

CLOSE ALL/DATABASE/INDEX

CONVERT

DELETE FILE

ERASE

INSERT

MODIFY STRUCTURE

PACK

RENAME

ZAP

Als een van deze commando's toch tijdens een transactie wordt gebruikt, wordt een foutmelding weergegeven.

De volgende commando's zijn toegestaan in transacties zolang ze geen geopende bestanden sluiten of bestaande bestanden overschrijven.

COPY TO <nieuw bestand> [STRUCTURE EXTENDED]

COPY STRUCTURE TO <nieuw bestand>

CREATE [FROM]

EXPORT TO <bestandsnaam>

IMPORT FROM <bestandsnaam>

INDEX...TO <nieuw bestand>

JOIN...TO <nieuw bestand>

SET CATALOG TO <nieuw bestand>

SET INDEX TO

SORT...TO <nieuw bestand>

TOTAL...TO <nieuw bestand>

USE

Wanneer door een commando wordt geprobeerd een bestaand bestand te overschrijven of een geopend database-bestand of indexbestand te sluiten, wordt een foutmelding weergegeven.

Het commando INDEX kan niet worden gebruikt als daarmee een bestaand .ndx-of .mdx-bestand of een bestaande .mdx-label wordt overschreven, of wanneer een willekeurig niet-bijbehorend indexbestand geopend is.

Het commando UNLOCK is een speciaal geval. Aangezien het commando geen foutmelding oplevert, heeft het geen effect gedurende een transactie.

Tip

Bestand- en recordvergrendeling die tijdens een actieve transactie zijn aangebracht, worden pas opgeheven wanneer de transactie is voltooid. In een multi-user systeem kunt u voorkomen dat gegevens voor lange tijd niet toegankelijk zijn door er voor te zorgen dat uw programma's het toevoegen van gegevens hebben afgesloten op het moment dat u begint met een transactie. Neem in een transactie alleen de noodzakelijke commando's op en rond de transactie zo snel mogelijk af, zodat de vergrendelingen kunnen worden opgeheven.

BEGIN/END TRANSACTION

Voorbeelden

De volgende routines, Factuur en Fout_verw, zijn voorbeelden van transactieprocedures met foutencontrole en -verwerking:

```
PROCEDURE Factuur
SET REPROCESS TO 15
ON ERROR DO Fout_verw
BEGIN TRANSACTION
    USE Factuur ORDER Rek_nr
    SCAN FOR .NOT. Factuur
        .
        .
        .
    ENDSCAN
END TRANSACTION
IF .NOT. ROLLBACK()
    @ 21, 15 SAY "ROLLBACK niet geslaagd; haal reservebestand op"
    @ 22,15 SAY "om door te gaan."
ENDIF
ON ERROR
RETURN
PROCEDURE Fout_verw
choice = " "
DO CASE
    CASE ERROR() = 108        && Bestand in gebruik door een andere gebruiker
        @ 21,15 SAY "Een van de bestanden is door een ander geopend."
        @ 22,15 SAY "Opnieuw proberen? (J/N)";
        GET choice
        READ
        @ 21, 15 CLEAR TO 22,65
        IF UPPER (choice) = "J"
            RETRY
        ELSE
            IF COMPLETED()
                *—- Geen fout tijdens transactie.
                @ 21,15 SAY "Bestanbd is al in gebruik."
                @ 22, 15 SAY "Terug naar hoofdmenu."
                RETURN TO MASTER
            ENDIF
            @ 21, 15 SAY "Bezig met ROLLBACK. Even wachten a.u.b..."
            ROLLBACK
        ENDIF
        RETURN TO MASTER
    CASE ERROR() = 109
    *** Soortgelijke routine.
ENDCASE
RETURN
```

Zie ook

CHANGE(), COMPLETED(), ISMARKED(), RESET, ROLLBACK, ROLLBACK()

BROWSE

BROWSE is een schermgroot, menugesteund commando voor het bewerken van en toevoegen van records aan database-bestanden (.dbf) en visies.

Syntaxis

BROWSE [NOINIT] [NOFOLLOW] [NOAPPEND] [NOMENU] [NOEDIT] [NODELETE] [NOCLEAR] [COMPRESS] [FORMAT] [LOCK <Nuitdr>] [WIDTH <Nuitdr>] [FREEZE <veldnaam>] [WINDOW <vensternaam>] [FIELDS <veldnaam 1> [/R] [/>kolombreedte>] / <naam rekenveld 1> [/>kolombreedte] = <uitdrukking 1> [,<veldnaam 2> [/R] [/>kolombreedte>] / <naam rekenveld 2> [/>kolombreedte>] = <uitdrukking 2>] ...]

Standaardinstellingen

Als er geen database-bestand of visie actief is als u het commando BROWSE geeft, wordt u gevraagd de naam van een database-bestand op te geven.

Voor rekenvelden geldt altijd alleen leesrecht.

Wanneer u het commando BROWSE geeft vanuit EDIT via **F2 Gegevens** of vanuit het Control Center, gebruikt BROWSE het actieve indelingsbestand (.fmt).

Gebruik

Met BROWSE kunt u records uit database-bestanden of visies in tabelvorm weergeven. Alle velden worden weergegeven in de volgorde die wordt bepaald door de bestandsstructuur of de visiedefinitie, of in de volgorde die u opgeeft bij de optie FIELDS.

Als het bestand is beveiligd met PROTECT, hebt u het recht de informatie uit de velden te lezen en te bewerken als u daartoe volgens uw toegangsniveau gerechtigd bent. Als uw toegangsniveau te laag is, wordt de melding **Geen toegangsrecht op dit niveau** weergegeven.

Met **F2 Gegevens** kunt u overschakelen van BROWSE naar EDIT. U kunt overschakelen naar het query-ontwerp door **Over naar query-ontwerp** te kiezen uit het menu **Stop** of door te drukken op **Shift-F2 Ontwerp**.

U kunt alle velden bewerken behalve rekenvelden of velden met alleen leesrecht. Wanneer u wijzigingen aanbrengt in een veld waarvan een rekenveld afhankelijk is, wordt dit rekenveld opnieuw berekend en weergegeven. Rekenvelden bestaan alleen tijdens een BROWSE-sessie, tenzij de velden zijn opgegeven bij het commando SET FIELDS TO.

Als een indexbestand actief is en u een sleutelveld bewerkt, krijgt het record een nieuwe plaats, afhankelijk van de nieuwe sleutelwaarde in de index.

U kunt ook records toevoegen aan het actieve bestand of de actieve visie. Verplaats de cursor naar het einde van het bestand en druk op *pijl omlaag*. U wordt gevraagd of u nieuwe records aan het bestand wilt toevoegen. Als u met J antwoordt, wordt de APPEND-modus geactiveerd en kunt u een record aan het bestand toevoegen. U kunt ook records toevoegen aan een op dat moment geselecteerd database-bestand dat in een visie is opgenomen.

Druk op **F10** om het balkmenu van BROWSE te activeren. De menu's van BROWSE worden uitvoerig beschreven in *dBASE IV voor beginners* en *Menu's van dBASE IV*. Raadpleeg deze boeken voor meer informatie.

Als u BROWSE vanaf de commandostip aanroept, keert u ook terug naar de commandostip wanneer u BROWSE verlaat. In programmabestanden keert u terug naar de opdracht die volgt op het commando BROWSE.

Opties

Met NOINIT kunt u de commandoregelopties die u hebt gebruikt bij een eerder gegeven BROWSE-commando opnieuw gebruiken bij het commando EDIT. NOINIT zorgt ervoor dat het commando BROWSE de BROWSE-tabel niet opnieuw initialiseert, maar dat in plaats daarvan de tabel van het laatste BROWSE-commando wordt gebruikt.

NOFOLLOW heeft alleen gevolgen voor geïndexeerde bestanden. Wanneer u een sleutelveld bewerkt, krijgt het record normaal gesproken een nieuwe plaats in de index en blijft dat record het actieve record. Als u het commando NOFOLLOW geeft voordat u de inhoud van een veld wijzigt, wordt het record opnieuw geïndexeerd, maar wordt het record dat de oude plaats van het opnieuw geïndexeerde record inneemt, het actieve record.

Met NOAPPEND voorkomt u dat nieuwe records kunnen worden toegevoegd aan het actieve bestand.

Met NOMENU voorkomt u dat het balkmenu wordt aangeroepen en weergegeven.

Met NOEDIT geeft u alle velden in de tabel het attribuut leesrecht. U kunt nog wel records aan het bestand toevoegen. U kunt records ook van een wismarkering voorzien.

Met NODELETE voorkomt u dat records kunnen worden gewist.

Met NOCLEAR blijft de BROWSE-tabel op het scherm staan als u BROWSE verlaat.

Met COMPRESS verkleint u de indeling van de tabel zodat twee regels meer op het scherm kunnen worden weergegeven. De titels van de kolommen worden op de bovenste regel van de tabelranden geplaatst en de blanco regel tussen de titels en de gegevens wordt verwijderd. Op een beeldscherm met 25 regels kan BROWSE normaal gesproken maximaal 17 records weergeven. Als u de optie COMPRESS gebruikt kunnen 19 records worden weergegeven. COMPRESS is niet te gebruiken in de modus EDIT.

Met FORMAT worden door BROWSE de opties van het commando @...GET gebruikt die zijn opgegeven in het actieve indelingsbestand (.fmt). Alle opties van het commando @...GET worden dan gebruikt, behalve de plaatsing van de velden op het scherm. Rij- en kolomcoördinaten worden genegeerd, omdat BROWSE de velden altijd in een tabel op het scherm weergeeft.

Wanneer er een indelingsbestand actief is, worden gegevens bewerkt aan de hand van de attributen van het database-bestand en de specificaties van het indelingsbestand. De optie FORMAT heeft een hogere priortiteit dan een mogelijke lijst die is opgegeven bij FIELDS. In dit geval bepaalt het indelingsbestand namelijk welke velden bij BROWSE worden gebruikt.

LOCK bepaalt hoeveel opeenvolgende velden links op het scherm blijven staan wanneer de BROWSE-tabel wordt verschoven. De waarde van <nuitdr> moet liggen tussen 0 en het totale aantal BROWSE-velden. Wanneer u op **F3** of **F4** drukt om velden naar links of rechts te verschuiven, verschuiven de velden die met LOCK zijn *vergrendeld*, niet. Met LOCK 0 worden alle vergrendelde velden weer ontgrendeld. In de modus EDIT wordt LOCK genegeerd.

Met WIDTH stelt u een maximale breedte in voor de kolommen van alle velden van de BROWSE-tabel. De waarde van <nuitdr> moet liggen tussen 4 en 99. De bij WIDTH opgegeven breedte heeft een hogere prioriteit dan de breedte die wordt bepaald door de structuur van de database. Als WIDTH en <kolombreedte> worden gebruikt voor een veld, wordt de laagste van de twee waarden gebruikt.

De optie WIDTH is niet van toepassing op memovelden of logische velden. Numerieke velden en datumvelden worden niet weergegeven als de kolom smaller is dan die van de velden in de database-structuur.

Met de optie WIDTH worden bij tekenvelden indien nodig zowel gegevens als de kolomtitel afgekapt. Bij andere gegevenstypen wordt alleen de kolomtitel afgekapt (gegevens die te lang zijn, worden gewoon niet weergegeven).

Met FREEZE worden uw activiteiten beperkt tot één veld of kolom in het bestand. Als het veld alleen kan worden gelezen, kunt u het niet bewerken. Andere velden die in de veldenlijst of de tabel van het database-bestand staan, worden wel weergegeven als deze op het scherm passen, maar kunnen niet worden bewerkt.

De optie FREEZE is niet van invloed op het commando EDIT. Als u naar de modus EDIT schakelt, is de cursor niet langer beperkt tot een speciaal veld. Zodra u teruggaat naar de modus BROWSE, is de optie FREEZE weer van kracht.

Met WINDOW activeert u een venster dat wordt gebruikt voor de BROWSE-tabel. De rest van het scherm blijft intact. U kunt zoomen tussen schermgrote weergave en weergave in het venster door op **F9** te drukken. Wanneer u BROWSE verlaat, wordt het venster automatisch gedeactiveerd. Als een venster actief is wanneer het commando BROWSE wordt gegeven, worden de gegevens door BROWSE in dat venster weergegeven.

Wanneer BROWSE in een venster wordt gebruikt en u zoomt tijdens het bewerken een memoveld met **F9**, dan wordt daardoor het eerste venster inactief gemaakt, waarna het beeldscherm in de modus van de schermgrote weergave blijft staan. Door vanuit een BROWSE-tabel in een alternatief venster memovelden te openen, wordt het oorspronkelijke BROWSE-venster inactief gemaakt en wordt alle informatie naar het venster met de memovelden verplaatst.

Met FIELDS kunt u velden kiezen en vervolgens opgeven in welke volgorde de velden in de tabel moeten worden opgenomen. Als het bestand of de visie is beveiligd met PROTECT en u geen toegang hebt tot een veld, wordt het betreffende veld niet weergegeven. Met de optie

FIELDS kunt u ook een veld van het attribuut leesrecht voorzien en rekenvelden maken en benoemen die in de BROWSE-tabel worden opgenomen.

Elk rekenveld is opgebouwd uit een toegewezen veldnaam en een uitdrukking die de waarde van het rekenveld bepaalt, bijvoorbeeld *Commissie =Percentage*Verkoopprijs*. De veldnaam die u toewijst wordt de titel van de kolom voor het rekenveld in de BROWSE-tabel. dBASE IV bepaalt de breedte van een rekenveld door de uitdrukking in het actieve record te evalueren of de grootst mogelijke waarde te berekenen.

Bij tekenvelden wordt de lengte ingesteld op de lengte van de uitkomst van het rekenveld. De minimale lengte die bij tekenvelden wordt gebruikt is 10. Bij logische velden wordt de lengte ingesteld op 1, bij datumvelden op 8 en bij numerieke velden op 20.

Met de optie /R voorziet u een veld van het attribuut leesrecht. De leesrechtvlag die is ingesteld met PROTECT heeft een hogere prioriteit dan deze optie. Rekenvelden hoeven niet met /R te worden beveiligd, omdat ze altijd het attribuut leesrecht hebben.

De optie <kolombreedte> is een numerieke constante die de kolombreedte van een veld bepaalt. De optie kan worden gebruikt voor alle velden in de veldenlijst. Op de commandoregel moet <kolombreedte> worden voorafgegaan door een schuine streep (/).

<kolombreedte> kan een waarde zijn van 4 tot en met 99 voor tekenvelden en een waarde van 8 tot en met 99 voor datumvelden en numerieke velden. <kolombreedte> wordt niet gebruikt voor memovelden en logische velden. <kolombreedte> wordt evenmin gebruikt als de breedte groter is dan de in WIDTH opgegeven waarde, of als de waarde kleiner is dan de breedte die in de structuur van het database-bestand voor datumvelden en numerieke velden is opgegeven. <kolombreedte> kan alleen worden gebruikt om tekenvelden smaller te maken. De optie kan niet worden gebruikt voor het smaller maken van kolomtitels.

Voorbeelden

In een programmabestand kunt u BROWSE gebruiken om maximaal zeven records uit het database-bestand Transact weer te geven. Verklein daarvoor de weergave tot zeven records door een venster met de naam Deel te definiëren. Zoek in het database-bestand de orders die in maart zijn geplaatst. De gebruiker moet de BROWSE-tabel kunnen doorlopen, maar mag geen wijzigingen in het database-bestand aanbrengen. Ten slotte moet de BROWSE-tabel op het scherm blijven staan terwijl het onderste deel van het beeldscherm wordt gebruikt door een andere routine. Als de tabel op het scherm blijft staan, kan de gebruiker de gegevens zien.

```
USE Transact
SET FILTER TO MONTH(Dat_trans) = 3        && Filter voor maart.
GO TOP                                    && Recordwijzer verplaatsen.
DEFINE WINDOW Deel FROM 2,10 TO 9,60
BROWSE NOAPPEND NOEDIT NODELETE NOCLEAR NOMENU COMPRESS WINDOW Deel
DO Vlg_prg
```

Het scherm ziet er na het BROWSE-commando uit het voorbeeld als volgt uit:

```
KLANT_NR  BESTEL_NR  DAT_TRANS  FACTUUR  TOTAAL_REK
L00001    87-109     09-03-87   W            415,00
C00002    87-110     09-03-87   W            175,00
L00002    87-111     11-03-87   O           1000,00
L00001    87-112     20-03-87   W            700,00
A00005    87-113     24-03-87   W            125,00
B12000    87-114     30-03-87   O            450,00

. use transact
. set filter to month(dat_trans)=3
. go top
TRANSACT: Record nr.      5
. define window partial from 2,10 to 9,60
. browse noappend noedit nodelete noclear nomenu compress window partial
Bladeren D:\...voorbld\TRANSACT  Rec 5/12  Bestan
```

Afbeelding 2-1 BROWSE-venster

<SCR>file:2-49.scr

Zie ook

APPEND, EDIT, PROTECT, SET MEMOWIDTH, SET REFRESH, SET WINDOW OF MEMO

CALCULATE

Met CALCULATE kunt u financiële en statistische functies op gegevens uitvoeren.

Syntaxis

CALCULATE [bereik] <optielijst>
[FOR <voorwaarde>] [WHILE <voorwaarde>]
[TO <lijst geheugenvariabelen>/TO ARRAY <reeksnaam>]

waarbij <optielijst> een van de volgende functies kan zijn:

AVG(<Nuitdr>)
CNT()
MAX(<uitdr>)
MIN(<uitdr>)
NPV(<disconto>,<flows>,<hoofdsom>)
STD(<Nuitdr>)
SUM(<Nuitdr>)
VAR(<Nuitdr>)

Standaardinstelling

Als u geen bereik opgeeft, worden alle records in het actieve bestand gebruikt.

Gebruik

Het commando CALCULATE berekent alle records in het actieve database-bestand totdat het gehele bereik is doorlopen, of de voorwaarde FOR of WHILE niet langer waar is. De uitkomst van het commando is één of meer getallen van het type N (vast) of het type F (zwevend), afhankelijk van het gegevenstype van het argument.

Als u de clausule FOR opgeeft, wordt elk record berekend en als het resultaat waar (.T.) is, worden de opgegeven functies berekend en uitgevoerd.

Als u de optie TO ARRAY opgeeft, worden de uitkomsten opgeslagen in de opgegeven reeks. De reeks moet ééndimensionaal zijn. De elementen van de reeks worden vanaf het eerste element gevuld.

Als SET TALK op ON is ingesteld, worden de uitkomsten op het scherm weergegeven. Als SET HEADING op ON is ingesteld, worden de uitkomsten ook voorzien van een label met de naam van de functie en de naam van het veld.

Opties

De financiële en statistische functies worden hierna beschreven. U kunt de uitkomsten van deze functies opslaan met de optie TO.

OPMERKING
Bij het berekenen van het maximum worden lege records gebruikt en deze worden als velden met de waarde nul behandeld.

AVG(<Nuitdr>). Hiermee berekent u het gemiddelde van de waarden in een bepaald veld van een database. De uitkomst is een getal dat van hetzelfde gegevenstype is als het argument.

Met CNT() telt u de records in een database-bestand. De voorwaarde FOR wordt voor ieder veld verwerkt. Als de voorwaarde waar is (.T.) wordt de telling met één verhoogd.

Met MAX(<uitdr>) wordt het hoogste getal in een opgegeven veld van een database bepaald. <uitdr> kan een numerieke uitdrukking, een datumuitdrukking of een tekenuitdrukking zijn. Normaal gesproken is de uitdrukking een veldnaam of heeft de uitdrukking betrekking op een veldnaam.

Met MIN(<uitdr>) wordt het laagste getal in een bepaald veld van een database bepaald. Deze optie werkt op dezelfde manier als MAX(), maar berekent de minimum waarde.

Met NPV(<disconto>, <flows>, <hoofdsom>) wordt de netto huidige waarde van het opgegeven veld van de database berekend.

<disconto> is het disconto weergegeven als decimaal getal.

<flows> is een reeks periodieke cashflow-waarden (voorzien van +/-). Als <hoofdsom> niet wordt gebruikt, begint de reeks met de lopende periode. <flows> kan elke geldige numerieke uitdrukking zijn, maar is gewoonlijk een veldnaam of een uitdrukking die betrekking heeft op een veldnaam.

<hoofdsom> is een numerieke uitdrukking die het investeringsbedrag weergeeft. De waarde moet een negatief getal zijn omdat deze waarde een negatieve cashflow inhoudt. De uitvoer van NPV() kan type N (vast) of type F (zwevend) zijn, afhankelijk van de argumenten die worden gebruikt.

Met STD(<Nuitdr>) wordt de standaarddeviatie van de elementen in het veld van een database berekend. De formule voor de standaarddeviatie is de wortel uit de variantie. <Nuitdr> is een numerieke uitdrukking en is gewoonlijk een veldnaam of een uitdrukking die betrekking heeft op een veldnaam. De uitvoer van STD() kan type N (vast) of type F (zwevend) zijn, afhankelijk van de argumenten die worden gebruikt.

Met SUM(<Nuitdr>) wordt de som van de waarden in een bepaald veld van een database berekend. De voorwaarde FOR wordt voor ieder veld berekend. Als de voorwaarde waar (.T.) is, wordt de waarde van dat record bij het totaal opgeteld. <Nuitdr> is een numerieke uitdrukking en is gewoonlijk een veldnaam of een uitdrukking die betrekking heeft op een veldnaam.

Met VAR(<Nuitdr>) wordt de populatievariantie van de waarden in een bepaald veld van een database berekend. <Nuitdr> is een numerieke uitdrukking en is gewoonlijk een veldnaam of een uitdrukking die betrekking heeft op een veldnaam. De uitvoer van VAR() is een getal van het type F (zwevend).

CALCULATE

Voorbeelden

Om het totaal van alle gefactureerde bestellingen in het database-bestand Transact te berekenen, gaat u als volgt te werk:

```
. USE Transact
. CALCULATE FOR Factuur SUM(Totaal_rek), CNT() TO totaal, cnt
        SUM(Totaal_rek) CNT()
            6965,00          8,00
. ? "De "+LTRIM(STR(cnt,3))+" bestellingen belopen ƒ"+LTRIM(STR(totaal,8,2))
De 8 bestellingen belopen ƒ6965,00
```

Om de gemiddelde rekening en de standaarddeviatie van het veld Totaal_rek te berekenen, gaat u als volgt te werk:

```
. USE Transact
. CALCULATE STD(Totaal_rek), AVG(Totaal_rek) TO Mdeviatie, Mgemiddeld
 12 records
      STD(Totaal_rek)     AVG(Totaal_rek)
            555,92                840
. Mdeviatie = LTRIM(STR(Mdeviatie,8,2))
555,92
. Mgemiddeld = LTRIM(STR(Mgemiddeld,8,2))
840,00
. ? "De gemiddelde rekening is ƒ"+Mgemiddelde+", marge van ƒ"+Mdeviatie+"."
De gemiddelde rekening is ƒ840,00, marge van ƒ555,92.
```

Om na te gaan wanneer de laatste transactie heeft plaatsgevonden en wat de grootste order tot nog toe is, gaat u als volgt te werk:

```
. USE Transact
. CALCULATE MAX(Dat_trans), MAX(Totaal_rek) TO laatste, grootste
       MAX(Dat_trans)    MAX(Totaal_rek)
10-04-87              1850,00
. SET CENTURY ON
. ? "De laatste transactie vond plaats op " + DMY(laatste) + "."
De laatste transactie vond plaats op 10 april 1987.
. ? "De grootste order bedroeg ƒ " + LTRIM(STR(grootste,8,2)) + "."
De grootste order bedroeg ƒ1850,00.
```

Zie ook

AVERAGE, COUNT, SUM

CALL

Met het commando CALL kunt u modules van binaire programmabestanden aanroepen die in het geheugen zijn geladen. U moet de binaire programmabestanden eerst met het commando LOAD hebben geladen.

Syntaxis

CALL <modulenaam> [WITH <uitdrukkingenlijst>]

Gebruik

Met het commando CALL wordt een binair programmabestand uitgevoerd dat met het commando LOAD in het geheugen is geladen. dBASE IV behandelt elk geladen bestand als een subroutine of module in plaats van als een extern programma (dat kan worden uitgevoerd met het commando RUN). Daarom hoeft u het programma niet telkens opnieuw te laden als u het wilt gebruiken, maar kan het gewoon in het geheugen opgeslagen blijven. Er kunnen maximaal 16 binaire programmabestanden tegelijkertijd in het geheugen zijn opgeslagen en elk bestand mag maximaal 32.000 bytes groot zijn.

Wanneer u een binair programmabestand met CALL aanroept, moet u de naam van het bestand zonder de extensie .bin opgeven. Als u het bestand aanroept, kunt u een tekenuitdrukking, een geheugenvariabele of een reekselement van een willekeurig type meezenden naar het binaire programmabestand. Alle geheugenvariabelen van het tekentype en alle uitdrukkingen eindigen met een nulteken (ASCII-waarde nul).

Wanneer de programmamodule wordt aangeroepen, wijst Code Segment (CS) naar het punt in het geheugen waar de module begint. Data Segment (DS) en BX-registers bevatten het adres van de eerste byte van de geheugenvariabele, het veld, het element van de reeks of van de tekenuitdrukking.

Opties

De optie WITH kan een tekenuitdrukking, veldnaam, geheugenvariabele of element van een reeks zijn. Er kunnen maximaal zeven uitdrukkingen in <uitdrukkingenlijst> worden gebruikt. Geheugenvariabelen, velden en elementen mogen van elk type zijn. Als u een uitdrukking meezendt, kunt u de waarde daarvan niet wijzigen omdat uitdrukkingen als constante waarden worden meegezonden. Als u een veld of een geheugenvariabele (of een reekselement) meezendt, kunt u de waarde daarvan wel wijzigen.

Voorbeeld

Zie het commando LOAD.

Zie ook

CALL(), LOAD, RELEASE, RUN/!

CANCEL

Met CANCEL wordt de uitvoering van een commandobestand beëindigd, worden alle geopende commandobestanden gesloten en keert u terug naar de commandostip. Procedurebestanden worden niet gesloten.

Syntaxis

CANCEL

Gebruik

Wanneer er in een dBASE-programma na het commando CANCEL nog tekst staat, wordt die tekst door dBASE IV genegeerd.

CANCEL wordt hoofdzakelijk gebruikt als u programma's van fouten wilt zuiveren.

Voorbeeld

Om het verwerken van commando's te beëindigen zodra een fout wordt ontdekt, kunt u CANCEL in een IF...ENDIF- of DO CASE...ENDCASE-structuur opnemen. In het volgende voorbeeld is de variabele Mfout al ingesteld:

```
IF Mfout        && Als Mfout waar is, wordt uitvoering
   CANCEL       && van commandobestand stopgezet.
ENDIF
```

Zie ook

DEBUG, DO, RESUME, RETRY, RETURN, SET REPROCESS, SUSPEND

CHANGE

CHANGE is een andere naam voor EDIT, een schermgroot commando waarmee u de inhoud van een record in het actieve database-bestand of de actieve visie kunt weergeven of bewerken.

Syntaxis

CHANGE [NOINIT] [NOFOLLOW] [NOAPPEND] [NOMENU] [NOEDIT] [NODELETE] [NOCLEAR] [<recordnummer>] [FIELDS <veldenlijst>] [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Gebruik

Het commando CHANGE is identiek aan het commando EDIT. Raadpleeg de bespreking van EDIT voor meer informatie over het gebruik van dit commando.

Zie ook

EDIT

CLEAR

Met CLEAR worden de gegevens van het scherm gewist, wordt de cursor naar de linker benedenhoek van het scherm verplaatst en worden alle nog uit te voeren GET-commando's die met het @-commando zijn gemaakt, opgeheven.

U kunt met verschillende vormen van het commando CLEAR ook database-bestanden sluiten, geheugenvariabelen, veldenlijsten, vensters, pop-up menu's en overige menu's opheffen en de typebuffer leegmaken.

Syntaxis

CLEAR [ALL/FIELDS/GETS/MEMORY/MENUS/POPUPS/SCREENS/TYPEAHEAD/
WINDOWS]

Gebruik

Bij het CLEAR-commando gebruikt u geen opties of slechts één optie. U mag niet meer dan één optie in een commandoregel gebruiken.

Opties

Met de optie ALL worden alle geopende database-bestanden gesloten, alle geheugenvariabelen (behalve de systeemvariabelen), reekselementen, pop-up menu- en overige menudefinities opgeheven en wordt werkgebied 1 gekozen. Door de database-bestanden te sluiten worden ook de bijbehorende indexbestanden (zowel .ndx als .mdx), indelingsbestanden (.fmt) en memobestanden (.dbt) gesloten. Met CLEAR ALL wordt ook het catalogusbestand (.cat) gesloten als dat actief is.

Met de optie FIELDS wordt de veldenlijst die met het commando SET FIELDS is gemaakt, opgeheven. CLEAR FIELDS wordt alleen uitgevoerd wanneer u SET FIELDS hebt gebruikt of wanneer het commando door een visiebestand of een query-bestand is geactiveerd. In tegenstelling tot SET FIELDS TO, waarmee alleen de actieve database-velden uit de veldenlijst worden verwijderd, worden met CLEAR FIELDS alle velden opgeheven, inclusief de velden uit andere werkgebieden. Door CLEAR FIELDS wordt automatisch het commando SET FIELDS OFF uitgevoerd.

Met de optie GETS worden alle @...GET-commando's na het laatste CLEAR ALL-, CLEAR GETS- of READ-commando opgeheven. Behalve wanneer de GETS-parameter met behulp van een Config.db-bestand opnieuw is gedefinieerd, kunt u in dBASE IV 128 @...GET-commando's opgeven voordat u het commando CLEAR GETS of READ moet gebruiken.

Het maximaal toegestane aantal GET-commando's is 1023. Het werkelijke aantal dat u kunt gebruiken, wordt echter beperkt door de hoeveelheid geheugenruimte van de computer (raadpleeg het boekje *Aan de slag met dBASE IV* voor informatie over de geheugenruimte). Voor de meeste doeleinden voldoet de standaardwaarde 128.

Met het commando READ heft u alle GET-commando's op, tenzij u de optie SAVE gebruikt. Als u de optie SAVE gebruikt, moet u het commando CLEAR GETS gebruiken voordat het maximum aantal GET-commando's is bereikt.

U heft met CLEAR GETS geen geheugenvariabelen of reekselementen op.

Met de optie MEMORY heft u alle geheugenvariabelen (behalve de systeemvariabelen) en reekselementen op. Wanneer u het commando CLEAR MEMORY vanaf de commandostip invoert, wordt hiermee dezelfde functie uitgevoerd als met het commando RELEASE ALL. Wanneer u CLEAR MEMORY echter in programma's gebruikt, heft u alle PUBLIC en PRIVATE geheugenvariabelen en elementen op, terwijl u met RELEASE ALL alleen de gedefinieerde PRIVATE geheugenvariabelen en elementen opheft in het programma dat op dat moment wordt uitgevoerd.

Met de optie MENUS wist u alle gebruikersmenu's van het scherm en verwijdert u deze uit het geheugen. Dit commando gebruikt u om alle menu's van het scherm te wissen en de geheugenruimte die door de menu's in beslag wordt genomen, vrij te maken voor andere bewerkingen.

Met de optie POPUPS wist u alle pop-up menu's van het scherm en verwijdert u deze uit het geheugen. Dit commando gebruikt u om alle pop-up menu's van het scherm te wissen en de geheugenruimte die door de pop-up menu's in beslag wordt genomen, vrij te maken. Tijdens de uitvoering van dit commando wordt ook het actieve pop-up menu met DEACTIVATE inactief gemaakt en worden alle ON SELECTION-commando's die bij de pop-up menu's horen, gewist.

Met de optie SCREENS wordt de geheugenruimte vrijgegeven die in beslag werd genomen door de schermen die u met het commando SAVE SCREENS TO in geheugenvariabelen hebt opgeslagen. CLEAR SCREENS heeft geen effect op de vensters die worden aangeroepen met het commando ACTIVATE SCREENS, en ook niet op de schermindelingsbestanden (.scr, .fmt, .fmo). Het verschil tussen CLEAR SCREENS en RELEASE SCREENS is dat u met CLEAR SCREENS alle schermgeheugenvariabelen uit het geheugen verwijdert, terwijl u met RELEASE SCREENS alleen benoemde schermen uit het geheugen verwijdert.

Met de optie TYPEAHEAD wordt de typebuffer leeggemaakt. U gebruikt CLEAR TYPEAHEAD wanneer u er zeker van wilt zijn dat zich in de typebuffer geen toetsaanslagen bevinden. Dit is bijzonder nuttig in programma's waar u de gebruiker gegevens wilt laten invoeren voordat het programma wordt voortgezet. Plaats dit commando bijvoorbeeld voor commando's als CHANGE, READ, WAIT of soortgelijke commando's zodat u er zeker van bent dat dBASE IV op de juiste tekens reageert.

Met de optie WINDOWS worden alle vensters van het scherm verwijderd en uit het geheugen gewist. Alle definities van vensters die u niet hebt opgeslagen voordat u dit commando gebruikt, gaan verloren. Als u de vensterdefinities in een schijfbestand opslaat, kunt u de vensters met RESTORE WINDOWS altijd weer terugzetten en met CLEAR WINDOWS snel van het scherm en uit het geheugen verwijderen.

Zodra u het commando CLEAR WINDOWS geeft, is het scherm weer volledig beschikbaar. Eventuele tekst die door de vensters werd bedekt, wordt weer zichtbaar.

Zie ook

@, @...FILL, @...TO, CLOSE, RELEASE SCREENS, RESTORE WINDOW, SAVE SCREENS, SAVE WINDOW, WAIT

CLOSE

Met CLOSE worden ALTERNATE-bestanden, database-bestanden, indelingsbestanden, indexbestanden (.ndx en .mdx) en procedurebestanden gesloten.

Syntaxis

CLOSE ALL/ALTERNATE/DATABASES/FORMAT/INDEXES/PROCEDURE

Gebruik

Met CLOSE ALL worden alle typen bestanden gesloten en wordt werkgebied 1 opnieuw gekozen.

Wanneer u een bepaald type bestand met CLOSE wilt sluiten, gebruikt u achter het commando een van de reeds vermelde sleutelwoorden voor het gewenste bestandstype.

Als er een catalogus is geopend, heeft CLOSE DATABASES geen invloed op werkgebied 10. U sluit hiermee alle database-bestanden, maar ook alle bijbehorende indexbestanden (.ndx en .mdx), memobestanden (.dbt) en indelingsbestanden (.fmt).

Als u alleen het actieve database-bestand met de bijbehorende bestanden wilt sluiten, gebruikt u in plaats van het CLOSE-commando het USE-commando.

Voorbeelden

U sluit alle geopende bestanden als volgt:

```
. CLOSE ALL
```

U sluit alle geopende database-bestanden, indexbestanden en indelingsbestanden als volgt:

```
. CLOSE DATABASES
```

Zie ook

CLEAR ALL, QUIT

COMPILE

Met COMPILE wordt een bestand met dBASE IV-broncodes ingelezen en wordt een uitvoerbaar objectcodebestand gemaakt.

Syntaxis

COMPILE <bestandsnaam>[RUNTIME]

Standaardinstellingen

OPMERKING
Alle objectcodebestanden van dBASE IV 1.0 moeten in dBASE IV 1.1 opnieuw worden gecompileerd als u ze in versie 1.1 wilt draaien. In de meeste gevallen vindt het opnieuw compileren automatisch plaats. Door de commando's waarmee objectbestanden van dBASE worden gedraaid, wordt de versie van het codebestand gecontroleerd en vervolgens opnieuw gecompileerd wanneer blijkt dat het codebestand met versie 1.0 is gemaakt en het bijbehorende bronbestand wordt aangetroffen. Door dBASE IV wordt in de actieve bestandsindex en alle bestandsindexen in het dBASE-pad gezocht naar de bronbestanden. Een bestand wordt niet opnieuw gecompileerd als het bronbestand een andere naam heeft gekregen of niet wordt aangetroffen, of als het codebestand met DBLINK is gemaakt.

Bij de bestandsnaam moet u ook de stationsaanduiding opgeven, behalve wanneer het bestand zich in het standaardstation bevindt of wanneer er een pad naar het bestand is opgegeven. Bij de bestandsnaam moet u tevens een pad opgeven als het bestand zich niet in de standaard-bestandsindex bevindt of als er met het commando SET PATH geen pad naar het bestand is ingesteld.

Met COMPILE kunt u alleen bronbestanden compileren. Dit commando zoekt naar een bestand met de extensie .prg, tenzij u in de commandoregel een andere extensie opgeeft. Bronbestanden bevatten commando's en functies van dBASE IV en hebben meestal de extensie .prg (programmabestand), .prs (SQL-programma), .fmt (indelingsbestand), .frg (gegenereerd rapportformulier), .lbg (gegenereerd etiket), .qbe (query-bestand) of .upd (mutatie-query).

Hoewel het bronbestand een willekeurige extensie kan hebben, schrijft COMPILE het objectbestand altijd met de extensie .dbo weg.

Gebruik

Objectbestanden bevatten dBASE-code in een vorm die alleen kan worden uitgevoerd. Met eerdere versies van dBASE kon u geen objectcodebestanden genereren. De bestanden die u met COMPILE hebt gemaakt, kunt u daarom niet in dBASE III PLUS of dBASE III gebruiken. Door objectbestanden kan dBASE IV veel sneller commando's uitvoeren dan met eerdere versies mogelijk was.

Een objectbestand kunt u niet wijzigen. Het broncodebestand kunt u wel wijzigen, maar u moet dan controleren of de wijzigingen in het broncodebestand door COMPILE naar het objectcodebestand zijn gecompileerd.

COMPILE

Wanneer u de optie RUNTIME in de commandoregel opneemt, zorgt u ervoor dat dBASE IV eventuele fouten of waarschuwingen die worden veroorzaakt door commando's die in RunTime niet zijn toegestaan, afgedrukt.

Wanneer u met DO een programmabestand uitvoert, wordt de broncode naar een objectcodebestand gecompileerd (als er nog geen .dbo-bestand bestaat) en wordt vervolgens de objectcode uitgevoerd. Met COMPILE kunt u het objectcodebestand genereren zonder de code uit te voeren.

Wanneer u een .prg-bestand wijzigt, wordt door de tekstverwerker van dBASE IV, die u via MODIFY COMMAND of het Control Center kunt gebruiken, het oude .dbo-bestand verwijderd. Met het commando DO wordt het .prg-bestand daarna opnieuw gecompileerd en een nieuw .dbo-bestand gemaakt voordat de procedure wordt uitgevoerd.

Als SET DEVELOPMENT op ON is ingesteld en u met een externe tekstverwerker de bestaande bronbestanden (.prg) van het programma bewerkt, wordt door DO de tijd en de datum van een .prg-bestand vergeleken met de tijd en de datum van het bijbehorende .dbo-bestand. Als het .dbo-bestand ouder is dan het .prg-bestand, wordt het .prg-bestand door DO opnieuw gecompileerd voordat het wordt uitgevoerd.

Als u de MODIFY COMMAND-tekstverwerker niet gebruikt en SET DEVELOPMENT niet op ON instelt, moet u de broncodebestanden opnieuw compileren nadat u deze hebt gewijzigd. Doet u dit niet, dan worden de wijzigingen niet naar het objectbestand weggeschreven.

Met COMPILE wordt elke regel van het broncodebestand op syntactische juistheid en op correcte besturingsstructuur gecontroleerd. Syntactische fouten en niet-toegestane besturingsstructuren (zoals het ontbreken van ENDDO en ENDIF) worden gemarkeerd. dBASE IV laat hierover meldingen zien tijdens de compilatie, op voorwaarde dat SET TALK op ON is ingesteld. Als er in de commandoregel echter een macro wordt aangetroffen, wordt deze uitgewerkt tijdens de uitvoering en bovendien op syntaxis gecontroleerd.

Aangezien COMPILE het objectbestand altijd met een .dbo-extensie wegschrijft, moet u de broncodebestanden unieke namen geven zodat een bestand dat eerder met COMPILE is gecompileerd niet door een ander bestand met dezelfde naam kan worden overschreven.

Zorg ervoor dat elk broncodebestand een unieke naam heeft. Als twee .prg-bestanden dezelfde naam maar verschillende extensies hebben, wordt bij de compilatie van het ene bestand het .dbo-bestand van het andere overschreven. Geef een .dbo-bestand overigens geen andere naam. Wijzig in plaats hiervan de naam van het broncodebestand en compileer dit opnieuw.

```
* Centraal.prg
? "CENTRAAL"
RETURN
```

Dit bestand wordt naar een .dbo-bestand gecompileerd en bevat één procedure, Centraal. Wanneer u DO CENTRAAL invoert, wordt deze naam gebruikt voor het lokaliseren van het .dbo-bestand en daarna voor het lokaliseren van de procedure binnen het .dbo-bestand.

Een bronbestand kan meer dan een procedure bevatten, bijvoorbeeld:

```
* Centraal.prg
? "CENTRAAL"
DO Perifeer
RETURN

PROCEDURE Perifeer
? "PERIFEER"
RETURN
```

Zoals u ziet, wordt alleen aan de code die aan het begin van een bestand wordt aangetroffen, de standaard-procedurenaam gegeven.

Elke procedure die in een actief .dbo-bestand wordt aangetroffen, kan door het DO-commando worden gebruikt. Als A.dbo B.dbo oproept en B.dbo C.dbo oproept, kunnen alle procedures die binnen A, B en C zijn gedefinieerd, door elke procedure in C worden gebruikt. Het commando SET PROCEDURE uit dBASE III en dBASE III PLUS wordt door dBASE IV ondersteund maar u hebt dit commando alleen nodig om de procedures in een bestand dat niet door DO <bestandsnaam> wordt geactiveerd, te kunnen gebruiken.

Tips

Door COMPILE worden alle commandosleutelwoorden in een vooraf opgegeven volgorde geplaatst voordat de objectcode wordt gegenereerd. Meestal is de volgorde dezelfde als in de modellen onder het kopje "Syntaxis" bij alle commando's en functies. De compilatietijd voor lange commando's kan worden verkort door de commando's in de volgorde van de syntaxismodellen in te voeren.

dBASE IV optimaliseert de uitdrukkingen tijdens de compilatie. Als u in de broncode constanten gebruikt, wordt de waarde in het objectcodebestand door de compiler berekend en opgeslagen. Voorbeeld:

x = 1 + 3 + 4

Deze uitdrukking wordt geoptimaliseerd en opgeslagen als

x = 8

De vergelijking van twee tekenreeksconstanten in een uitdrukking kan problemen veroorzaken. Voorbeeld:

"abcde" = "abcd"

Deze uitdrukking resulteert in waar (.T.) als SET EXACT op OFF is ingesteld, maar in onwaar (.F.) als SET EXACT op ON is ingesteld. Als u SET EXACT tijdens de uitvoering van de code op ON instelt, is de uitdrukking onwaar omdat deze tijdens de compilatie als logisch onwaar (.F.) is geoptimaliseerd en opgeslagen.

Zie ook

CANCEL, DEBUG, DO, FUNCTION, MODIFY COMMAND, PROCEDURE, SET DEBUG, SET DEVELOPMENT, SET EXACT, SET PATH, SET PROCEDURE, SET TRAP, SUSPEND

CONTINUE

Met CONTINUE wordt gezocht naar het volgende record in het actieve database-bestand dat aan de voorwaarde voldoet die met het laatste LOCATE-commando is opgegeven.

Syntaxis

CONTINUE

Gebruik

Het zoeken met CONTINUE wordt beëindigd wanneer een record is gevonden dat aan de opgegeven LOCATE-voorwaarde voldoet of wanneer het einde van het LOCATE-*bereik* of het einde van het bestand is bereikt.

LOCATE en CONTINUE hebben alleen betrekking op het werkgebied waarin u deze commando's gebruikt. In elk werkgebied kunt u verschillende LOCATE- en CONTINUE-commando's gebruiken. Als u een werkgebied verlaat en vervolgens daarnaar terugkeert, is de LOCATE-voorwaarde nog steeds geldig.

Recordwijzer

Als SET TALK op ON is ingesteld en als door CONTINUE nog een record wordt gevonden dat aan de voorwaarde voldoet, wordt het recordnummer weergegeven. Wanneer dit niet het geval is, verschijnt de melding Einde van LOCATE-bereik, is het resultaat van FOUND() logisch onwaar (.F.) en wordt de recordwijzer bij het laatste record van het LOCATE-bereik of aan het einde van het bestand geplaatst. Als de recordwijzer zich aan het einde van het bestand bevindt, is het resultaat van EOF() logisch waar (.T.).

Voorbeeld

In het volgende voorbeeld worden records met "EIKENHOUT" in het veld Omschrijv van het database-bestand Magazijn gelokaliseerd:

```
. USE Magazijn
. LOCATE FOR "EIKENHOUT" $ Omschrijv
Record =          6
. CONTINUE
Record =          15
. CONTINUE
Einde van LOCATE-bereik
```

De melding **Einde van LOCATE-bereik** duidt aan dat er geen records meer zijn die aan de FOR- of WHILE-voorwaarde voldoen of zich binnen het bereik van het LOCATE-commando bevinden.

Zie ook

EOF(), FIND, FOUND(), LOCATE, SEEK, SEEK()

CONVERT

Met CONVERT wordt aan de structuur van een database-bestand een veld toegevoegd waarin gegevens zijn opgeslagen die nodig zijn voor de opsporing van multi-user vergrendelingen.

Syntaxis

CONVERT [TO <Nuitdr>]

Gebruik

Met dit commando wordt aan de structuur van het actieve database-bestand een tekenveld met de naam _dbaselock toegevoegd. De lengte van het veld wordt bepaald door de numerieke uitdrukking (een getal van 8 tot 24). De standaardinstelling is 16.

In het veld _dbaselock is voor de volgende waarden een gebied gereserveerd:

Teller = een hexadecimaal getal van twee bytes dat door de functie CHANGE() wordt gebruikt

Tijd = een hexadecimaal getal van drie bytes waarmee wordt vastgelegd op welke tijd de vergrendeling plaatsvond

Datum = een hexadecimaal getal van drie bytes waarmee wordt vastgelegd op welke datum de vergrendeling plaatsvond

Naam = een notatie in nul tot 16 tekens van de aanmeldingsnaam van de computer die de vergrendeling deed plaatsvinden (als er een vergrendeling actief is)

De eerste acht tekens van het veld zijn altijd voor de gedeelten teller, tijd en datum.

Als u het veld _dbaselock met CONVERT naar de standaardinstelling van 16 tekens converteert, wordt de lengte van de aanmeldingsnaam acht tekens. Als u het veld met CONVERT naar het maximum aantal van 24 tekens converteert, wordt de lengte van de aanmeldingsnaam 16 tekens. Als u het veld met CONVERT naar acht tekens converteert, wordt er geen ruimte voor de aanmeldingsnaam gereserveerd en wordt de naam niet in het record geschreven.

Steeds wanneer een record is bijgewerkt, wordt het gedeelte teller van _dbaselock opnieuw weggeschreven. Als u de functie CHANGE() gebruikt, wordt het gedeelte teller weer uit het geheugen ingelezen en vergeleken met de vorige waarde die in het geheugen werd opgeslagen toen het record voor de eerste maal werd ingelezen. Als de waarden verschillend zijn, heeft een andere gebruiker het record gewijzigd en is het resultaat van de functie CHANGE() logisch waar (.T.).

U kunt de waarde op onwaar (.F.) instellen door de recordwijzer te verplaatsen. Met GOTO RECNO() wordt het veld _dbaselock van het actieve record opnieuw ingelezen en is het resultaat van een daarop volgend CHANGE()-commando onwaar (.F.), tenzij een andere gebruiker in de tussentijd nog een wijziging heeft aangebracht.

CONVERT

De functie LKSYS() resulteert in de aanmeldingsnaam, de datum en de tijd uit het veld _dbaselock. Hiermee wordt aangeduid wie het record of het bestand heeft vergrendeld en wanneer de vergrendeling plaatsvond. Wanneer u een bestand vergrendelt, bevat het veld _dbaselock van het eerste record in het database-bestand de gegevens die worden gebruikt door CHANGE() en LKSYS().

Speciaal geval

In een netwerkomgeving moet het database-bestand in de modus EXCLUSIVE zijn geopend voordat u het commando CONVERT gebruikt.

Tip

Met CONVERT wordt het .dbf-bestand naar een nieuw bestand met de extensie .cvt gekopieerd. Vervolgens wordt een nieuw .dbf-bestand dat het veld _dbaselock bevat, gemaakt. Het .cvt-bestand bevat de bestandsstructuur die oorspronkelijk aanwezig was voordat CONVERT werd uitgevoerd.

Zie ook

CHANGE(), FLOCK(), LKSYS(), LOCK(), NETWORK(), SET LOCK, SET REPROCESS, UNLOCK

COPY

Met COPY wordt een actief database-bestand geheel of gedeeltelijk gekopieerd zodat een nieuw bestand wordt gemaakt. COPY is tevens het belangrijkste commando voor het exporteren van gegevens naar andere programma's dan dBASE.

Syntaxis

COPY TO <bestandsnaam>
[[TYPE] <bestandstype>] [FIELDS <lijst velden>]
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstellingen

Met dit commando worden alle records, inclusief de records met een wismarkering, gekopieerd tenzij SET DELETED op ON is ingesteld of tenzij u een bereik, FOR-clausule of WHILE-clausule hebt opgegeven om de te kopiëren records te begrenzen. Alle velden worden gekopieerd tenzij u bij FIELDS een lijst met velden hebt opgegeven of het commando SET FIELDS hebt gebruikt. Memovelden worden alleen gekopieerd als het nieuwe bestand ook een database-bestand van dBASE IV is.

Als u geen bestandstype samen met het commando invoert, wordt het bestand naar een database-bestand (.dbf) van dBASE IV gekopieerd.

Gebruik

De optionele sleutelwoorden [FIELDS <lijst velden>] kunt u alleen gebruiken met de database-bestanden in dBase IV en kopiëren het bestand production.mdx. U kunt het sleutelwoord TYPE niet gebruiken met de sleutelwoorden [FIELDS <lijst velden>] om een ander bestandstype op te geven.

Alleen met deze optie kunt u een indexbestand kopiëren.

Om met deze optie te kunnen werken, moet u over een leeg werkgebied beschikken.

De indexsleutels en de FOR-clausule in het gekopieerde MDX-bestand moeten in het gekopieerde database-bestand aanwezig zijn. Alle indirecte verwijzingen naar bestanden (aliassen) of de geheugenvariabelen die in het indexbestand worden gebruikt, moeten in het geheugen aanwezig zijn, zodat de gekopieerde index uitgevoerd kan worden

Als het bestand waarnaar (TO) wordt gekopieerd, een ASCII-tekstbestand of een bestand van een ander programma is, geeft u wel een van de bestandstype-opties op.

U kunt voor <bestandsnaam> een indirecte verwijzing gebruiken. Een indirecte verwijzing is een tekenuitdrukking die resulteert in een bestandsnaam en die u overal kunt gebruiken waar u een bestandsnaam moet opgeven. In deze uitdrukking moet u een operator gebruiken (meestal haakjes), zodat dBASE weet dat de tekenreeks een uitdrukking en geen letterlijke bestandsnaam is. Een indirecte verwijzing gebruikt u op dezelfde manier als het macrosubstitutie-teken, maar de indirecte verwijzing wordt veel sneller uitgevoerd.

Het veld _dbaselock in bestanden die met CONVERT zijn geconverteerd, wordt niet naar het nieuwe bestand gekopieerd. Als PROTECT is gebruikt, worden de velden waartoe de

gebruiker die het bestand kopieert geen toegang heeft (omdat de betreffende rechten niet zijn verleend), niet naar het nieuwe bestand gekopieerd.

OPMERKING
Door het COPY-commando wordt niet gecontroleerd of de bestanden die u samenstelt, compatibel zijn met andere programma's. Het is mogelijk dat u veldbreedten, recordlengten, veldaantallen of recordaantallen opgeeft die niet compatibel zijn met andere programma's. Noteer welke eisen andere programma's aan bestanden stellen voordat u met COPY database-bestanden exporteert.

Opties

Voor de bestandstypen die u exporteert, bestaan de volgende opties:

- DELIMITED - ASCII-bestand met scheidingstekens. De gegevens worden vanaf de linker kant teken voor teken naar dit bestand gekopieerd. Elk record wordt met een terugloop en een regeldoorvoer beëindigd. De velden worden door komma's gescheiden en bovendien staat elk tekengegeven tussen dubbele aanhalingstekens tenzij u een ander scheidingsteken opgeeft. Dit komt overeen met DELIMITED WITH ".
- DELIMITED [WITH <scheidingsteken>] - ASCII-tekstbestand (.txt) met komma's als veldscheidingstekens, waarbij de tekenvelden tussen scheidingstekens staan. Alle velden worden door komma's gescheiden. Het standaard-scheidingsteken voor tekenvelden is het dubbele aanhalingsteken tenzij u met de optie WITH <scheidingsteken> een ander scheidingsteken opgeeft. De records in het tekstbestand hebben een variabele lengte en worden allemaal met een terugloop en een regeldoorvoer beëindigd.
- DELIMITED [WITH BLANK] - ASCII-tekstbestand (.txt) met één spatie als veldscheidingsteken, waarbij de tekenvelden echter niet tussen scheidingstekens staan. De records in het tekstbestand hebben een variabele lengte en worden allemaal met een terugloop en een regeldoorvoer beëindigd.
- SDF - ASCII-bestand (.txt) met de indeling van de systeemgegevens (System Data Format). Er staan geen scheidingstekens tussen de tekengegevens en de velden worden niet door een teken gescheiden. De records hebben een vaste lengte die gelijk is aan de lengte van het record in het database-bestand. Elk record wordt met een terugloop en een regeldoorvoer beëindigd.
- DBASEII - dBASE II database-bestand (.db2). Het dBASE II-bestand krijgt in plaats van .dbf de extensie .db2 zodat dit van het oorspronkelijke dBASE IV-bestand kan worden onderscheiden. Voordat u dit bestand in dBASE II kunt gebruiken, moet u de bestandsnaam de extensie .dbf geven.
- DBMEMO3 - dBASE III PLUS-indeling voor database-bestanden (.dbf) en memoveldbestanden (.dbt). Als een database-bestand en het bijbehorende memobestand in dBASE IV zijn gemaakt of gewijzigd, kunt u deze niet in dBASE III PLUS openen. U kunt deze bestanden echter met COPY en TYPE DBMEMO3 kopiëren en daarna de kopieën in dBASE III PLUS openen. Alle numerieke velden van het type F (zwevend) die u met de optie DBMEMO3 kopieert, worden naar het type N (vast) geconverteerd.

- RPD - RapidFile-gegevensbestand (.rpd)
- FW2 - Framework II-database (.fw2)
- SYLK - MultiPlan spreadsheet-formule (.slk). De database-records worden naar MultiPlan-rijen en de database-velden naar MultiPlan-kolommen geconverteerd. Achter de naam van het uitvoerbestand komt geen extensie.
- DIF - VisiCalc versie 1-bestandsindeling (.dif). De database-records worden naar VisiCalc-rijen en de database-velden naar VisiCalc-kolommen geconverteerd.
- WKS - Lotus 1-2-3 spreadsheet-indeling (.wks) van versie 1A. De database-records worden naar Lotus 1-2-3 rijen en de database-velden naar Lotus 1-2-3 kolommen geconverteerd.

Wanneer u COPY gebruikt voor het wegschrijven naar een van de drie ondersteunde spreadsheet-indelingen (SYLK, DIF, WKS), worden de veldnamen als kolomkopregels in het resulterende bestand weggeschreven. Het bestand wordt in de volgorde van de rijen gemaakt.

Speciale gevallen

U converteert bestanden naar PFS:FILE met het commando EXPORT en niet met het commando COPY. Met EXPORT en COPY kunt u bestanden naar de bestandsindelingen van Framework II, dBASE II en RapidFile converteren. Het commando COPY kan echter geen PFS:FILE-formulieren maken.

Tips

Gebruik als naam van het database-bestand niet de letters A tot en met J of de letter M wanneer u met COPY TO naar een database-bestand van dBASE IV kopieert. Deze letters zijn gereserveerd voor de standaard-aliasnamen. U kunt echter wel bijvoorbeeld AA als naam van het database-bestand opgeven.

Als er een relatie actief is en u hebt met het commando SET FIELDS of met de FIELDS-clausule velden uit een ander werkgebied opgegeven, bevat het resulterende bestand de gegevens uit de gerelateerde records in de andere werkgebieden.

COPY

Voorbeeld

In het volgende voorbeeld worden alle records in het database-bestand Transact waarvan het Klant_nr C00002 is, naar een database-bestand met de naam Tijdl gekopieerd:

```
. USE Transact
. COPY TO Tijdl FOR Klant_nr = 'C00002'
                    2 records gekopieerd
. USE Tijdl.dbf
. LIST
Recordnr.   KLANT_NR    BESTEL_NR    DAT_TRANS    FACTUUR    TOTAAL_REK
        1   C00002      87-107       12-02-87     .T.         1250,00
        2   C00002      87-110       09-03-87     .T.          175,00
```

Zie ook

APPEND FROM, COPY FILE, COPY STRUCTURE, EXPORT, IMPORT, SET DELETED, SET FIELDS, SET SAFETY

COPY FILE

Met COPY FILE wordt van een willekeurig bestand een kopie gemaakt.

Syntaxis

COPY FILE <bestandsnaam> TO <bestandsnaam>

Gebruik

U moet voor beide bestanden de bestandsnamen en de bijbehorende extensies opgeven. Als u een bestand naar een ander station of een andere bestandsindex wilt kopiëren, moet u tevens de stationsaanduiding en de naam van de bestandsindex opgeven. De bestandsindex of de stationsaanduiding van het eerste bestand hoeft u niet op te nemen wanneer deze al met het commando SET PATH of met PATH-instelling in het bestand Config.db zijn vastgesteld.

U kunt met COPY FILE geen geopend bestand kopiëren.

Tips

Als u een database-bestand met memovelden of een bijbehorend mdx-bestand kopieert, moet u het bijbehorende memobestand (.dbt) en het bijbehorende indexbestand (.mdx) afzonderlijk kopiëren. U kunt met het COPY-commando records uit een geopend database-bestand kopiëren.

Gebruik als naam van het database-bestand niet de letters A tot en met J of de letter M aangezien deze letters zijn gereserveerd voor de standaard-aliasnamen. U kunt echter wel bijvoorbeeld AA als naam van het database-bestand opgeven.

Voorbeeld

In het volgende voorbeeld wordt van een dBASE-programmabestand een kopie gemaakt:

```
. COPY FILE Rekng.prg TO Oudrekng.prg
2123 bytes gekopieerd
```

Zie ook

COPY, SET PATH

COPY INDEXES

Met COPY INDEXES wordt een lijst met indexbestanden (.ndx) naar labels in één .mdx-bestand (meerdere indexlabels) geconverteerd.

Syntaxis

COPY INDEXES <lijst .ndx-bestanden> [TO <.mdx-bestand>]

Gebruik

Als u met de TO-clausule geen .mdx-bestand opgeeft, wordt de label naar het bijbehorende .mdx-bestand weggeschreven. Als er geen bijbehorend .mdx-bestand bestaat, wordt dit onder dezelfde naam als die van het actieve database-bestand gemaakt en wordt de aanhef van het database-bestand bijgewerkt ter aanduiding van de aanwezigheid van een bijbehorend .mdx-bestand.

Als u een TO-clausule gebruikt, wordt de label naar het opgegeven .mdx-bestand weggeschreven. Als het .mdx-bestand dat u in de TO-clausule opgeeft, niet bestaat, wordt een nieuw .mdx-bestand met de in de TO-clausule opgegeven bestandsnaam gemaakt.

De .ndx-bestanden moeten zijn geopend voordat u het commando COPY INDEXES gebruikt. U kunt met één COPY INDEXES-commando maximaal 10 indexbestanden (.ndx) naar labels in een .mdx-bestand kopiëren omdat in een werkgebied maximaal 10 indexbestanden geopend kunnen zijn.

Speciale gevallen

In een netwerk moet u het exclusieve gebruik van het database-bestand hebben voordat u een .ndx-bestand naar een .mdx-label kopieert.

Voorbeeld

In het volgende voorbeeld wordt het indexbestand Klntnaam van het database-bestand Afnemers naar een label in het bijbehorende .mdx-bestand geconverteerd:

```
. USE Afnemers INDEX Klntnaam
.COPY INDEXES Klntnaam
100% geïndexeerd  8 records geïndexeerd
. DISPLAY STATUS
Geselecteerde database:
Gebied:          1, geopende database: C:\DBASE\AFNEMERS.DBF Alias: AFNEMERS
                 Indexbestand:    C:\DBASE\KLNTNAAM.NDX Sleutel:Achternaam+Voornaam
Bijbehorend      MDX-bestand:     C:\DBASE\AFNEMERS.MDX
                 Indexlabel:      KLANTNAAM Sleutel: KLANTNAAM
                 Indexlabel:      KLANT_NR Sleutel: KLANT_NR
                 Indexlabel:      KLNTNAAM Sleutel: Achternaam+Voornaam
                 Memobestand:     C:\DBASE\AFNEMERS.DBT
```

Zie ook

COPY TAG, INDEX, KEY(), MDX(), NDX(), SET EXCLUSIVE, SET INDEX, SET ORDER, TAG(), USE

COPY MEMO

Met COPY MEMO worden de gegevens uit één memoveld naar een extern bestand gekopieerd.

Syntaxis

COPY MEMO <naam memoveld> TO <bestandsnaam> [ADDITIVE]

Standaardinstelling

Als u geen extensie opgeeft voor het bestand waarnaar (TO) wordt gekopieerd, wordt het bestand met de extensie .txt weggeschreven.

Gebruik

Met dit commando worden de gegevens uit een memoveld in het actieve record naar een opgeslagen bestand geëxporteerd.

Als u een bestand naar een ander station of een andere bestandsindex wilt kopiëren, moet u de stationsaanduiding en de naam van de bestandsindex opgeven.

U kunt de veldnaam door een alias laten voorafgaan en memovelden uit andere werkgebieden kopiëren. De alias hoeft u niet te gebruiken als het veld in een SET FIELDS-lijst is opgenomen.

Opties

Met de optie ADDITIVE wordt de inhoud van het memoveld aan het einde van het benoemde bestand toegevoegd. Als u de optie ADDITIVE niet gebruikt en de bestandsnaam al bestaat, overschrijft de inhoud van het memoveld eventuele gegevens in dat bestand.

Als SET SAFETY op OFF is ingesteld, worden de gegevens met dit commando zonder waarschuwing naar een bestand weggeschreven. Als SET SAFETY op ON is ingesteld en er een bestand met dezelfde naam in de doel-bestandsindex bestaat, verschijnt er eerst een melding op het scherm voordat het bestand wordt overschreven.

Voorbeeld

In het volgende voorbeeld worden de gegevens in het veld Overzicht naar een bestand met de naam Klnttkst weggeschreven. Hierbij worden de gegevens in het actieve record aan het einde van de reeds bestaande gegevens in het bestand toegevoegd:

```
. COPY MEMO Overzicht TO Klnttkst ADDITIVE
```

Zie ook

APPEND MEMO, COPY, COPY FILE

COPY STRUCTURE

Met COPY STRUCTURE wordt de structuur van een actief .dbf- of .dbt-bestand naar een nieuw bestand gekopieerd. Er worden geen records gekopieerd.

Syntaxis

COPY STRUCTURE TO <bestandsnaam> [FIELDS <lijst velden>]
[[WITH] PRODUCTION]

Standaardinstellingen

Bij de bestandsnaam moet u de stationsaanduiding en de bestandsindex opgeven als u het resulterende bestand niet naar het standaardstation of de standaard-bestandsindex wilt wegschrijven. Het bestand waarnaar (TO) wordt gekopieerd, krijgt een .dbf-extensie toegewezen, tenzij u een andere extensie opgeeft.

Gebruik

Als u de optie [WITH] PRODUCTION gebruikt, wordt met het commando COPY STRUCTURE een nieuw, leeg .mdx-bestand gemaakt met daarin alle uitdrukkingen van de indexlabels die in het nieuwe .dbf-bestand zijn opgenomen.

Als u de optie [WITH] PRODUCTION kiest, moet u over een vrij werkgebied beschikken.

De sleuteluitdrukkingen en FOR-clausules uit het gekopieerde bijbehorende .mdx-bestand moeten aanwezig zijn in het gekopieerde database-bestand. Alle indirecte bestandsverwijzingen (aliasnamen) en geheugenvariabelen die in het .mdx-bestand worden gebruikt, moeten in het geheugen staan om de gekopieerde index te kunnen laten functioneren.

Met COPY STRUCTURE wordt de gehele structuur van het database-bestand gekopieerd, tenzij u deze met de optie FIELDS of SET FIELDS beperkt. Het resultaat is een ander database-bestand met een identieke structuur of, als u dat bij de optie FIELDS opgeeft, met een deelverzameling van de velden uit het eerste bestand.

Als SET SAFETY op OFF is ingesteld, overschrijft een nieuw database-bestand een bestaand database-bestand zonder waarschuwing.

Het veld _dbaselock uit bestanden die met CONVERT zijn geconverteerd, wordt niet gekopieerd naar het nieuwe bestand.

Als PROTECT in gebruik is, worden de velden waarvoor de gebruiker, die het bestand kopieert, geen rechten bezit niet naar het nieuwe bestand gekopieerd.

Tip

U kunt met COPY STRUCTURE een structuur onder programmabesturing kopiëren zodat u tijdelijke database-bestanden maakt. U kunt dan met het commando APPEND records aan het transactiebestand toevoegen en met het commando REPLACE gegevens in de records plaatsen.

COPY STRUCTURE

Voorbeeld

In het volgende voorbeeld wordt de structuur van het database-bestand Afnemers naar een bestand met de naam Tijdl gekopieerd:

```
. USE Afnemers
. COPY STRUCTURE TO Tijdl
```

Zie ook

APPEND, APPEND BLANK, APPEND FROM, DISPLAY STRUCTURE, REPLACE, SET SAFETY

COPY STRUCTURE EXTENDED

Met COPY STRUCTURE EXTENDED wordt een nieuw database-bestand gemaakt waarvan de records de structuur van het actieve bestand bevatten.

Syntaxis

COPY TO <bestandsnaam> STRUCTURE EXTENDED

Gebruik

Met dit commando wordt een database-bestand met vijf velden gemaakt: FIELD_NAME, FIELD_TYPE, FIELD_LEN, FIELD_DEC en FIELD_IDX. De records in het nieuwe bestand bevatten de veldnaam, het gegevenstype, de veldbreedte, het aantal decimale posities (in een numeriek veld) en de indexlabel voor elk veld in het actieve database-bestand. Alleen die indexlabels waarbij tijdens de samenstelling van de structuur van het database-bestand "Ja" is opgegeven, worden naar het veld FIELD_IDX gekopieerd. Dit zijn enkelvoudige veldlabels die dezelfde naam hebben als het indexveld. Samengestelde indexen worden niet gekopieerd.

Dit commando wordt het meest gebruikt in toepassingsprogramma's in combinatie met het commando CREATE FROM. Met CREATE FROM wordt een database-bestand uit het uitgebreide structuurbestand gemaakt. U kunt op deze manier een database-bestand in een programma maken zonder de interactieve commando's CREATE of MODIFY STRUCTURE te hoeven gebruiken.

Het veld _dbaselock in bestanden die met CONVERT zijn geconverteerd, wordt niet naar het nieuwe bestand gekopieerd. Als PROTECT is gebruikt, worden de velden waartoe de gebruiker die het bestand kopieert geen toegang heeft (omdat de betreffende rechten niet zijn verleend), niet naar het nieuwe bestand gekopieerd.

Aangezien uitgebreide structuurbestanden die in dBASE IV zijn gemaakt, een FIELD_IDX-veld bevatten, zijn deze niet compatibel met dBASE III PLUS. In de structuren van de database-bestanden die met eerdere versies van dBASE zijn gemaakt, werden geen indexlabels opgenomen.

Voorbeeld

Zie het commando CREATE FROM.

Zie ook

APPEND FROM, COPY STRUCTURE, CREATE FROM, LIST/DISPLAY STRUCTURE

COPY TAG

Met COPY TAG worden labels van indexbestanden met meerdere indexen (.mdx) naar .ndx-indexbestanden geconverteerd.

Syntaxis

COPY TAG <label> [OF .mdx-bestandsnaam] TO <.ndx-bestandsnaam>

Standaardinstelling

Met de OF-clausule kunt u het .mdx-bestand dat de label bevat, opgeven.

Gebruik

Het database-bestand moet in gebruik zijn omdat met COPY TAG het .ndx-bestand op basis van de uitdrukking in de .mdx-bestandslabel opnieuw wordt gemaakt. De indexlabel die wordt gekopieerd, moet afkomstig zijn uit een geopend .mdx-bestand. Per keer kan slechts één label worden gekopieerd.

COPY TAG houdt rekening met de grens van maximaal 10 geopende ndx-bestanden voor het actieve .dbf-bestand. Wanneer u een nieuw .ndx-bestand maakt, kunnen er minder dan 10 .ndx-bestanden zijn geopend.

Voorbeeld

In het volgende voorbeeld wordt de label Bestel_nr uit het .mdx-bestand Magazijn naar het .ndx-bestand Art_nr gekopieerd:

```
. USE Magazijn
. COPY TAG Bestel_nr TO Art_nr.ndx
  100% geïndexeerd 17 records geïndexeerd
. SET INDEX TO Art_nr
  Hoofdindex: ART_NR
. DISPLAY STATUS
Geselecteerde database:
Gebied:   1, geopende database:  C:\SET\MAGAZIJN.DBF   Alias:        MAGAZIJN
          C:\SET\ART_NR.NDX Sleutel:        BESTEL_NR
          Hoofdindexbestand:
Bijbehorend    MDX-bestand:     C:\SET\MAGAZIJN.MDX
          Indexlabel:     BESTEL_NR          Sleutel: BESTEL_NR
          Indexlabel:     ARTIKEL            Sleutel: ARTIKEL
```

Zie ook

COPY INDEX, INDEX, MDX(), NDX(), SET INDEX, SET ORDER, TAG()

COPY TO ARRAY

Met COPY TO ARRAY wordt een bestaande reeks gevuld met de inhoud van een of meer records uit het actieve database-bestand.

Syntaxis

COPY TO ARRAY <naam reeks> [FIELDS <lijst velden>]
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Gebruik

Met dit commando worden geselecteerde records en velden uit een database-bestand naar een bestaande reeks gekopieerd.

Van elk record in het database-bestand wordt het eerste veld in de eerste kolom opgeslagen, het tweede veld in de tweede kolom, enzovoort. Elk record wordt een rij in de reeks. Dit proces gaat net zo lang door totdat er geen velden of geen reekskolommen meer zijn. Als het database-bestand meer velden bevat dan in de reeks kunnen worden opgeslagen, wordt het teveel aan velden niet opgeslagen. Als het database-bestand minder velden heeft dan de reeks, blijven de reekselementen ongewijzigd. Memovelden kunnen niet naar een reeks worden gekopieerd.

Als u een eendimensionale reeks definieert, zoals

```
. DECLARE Transact[5]
```

kunt u met COPY TO ARRAY alleen de eerste vijf velden van één record naar de reeks kopiëren. Om meer dan één record naar een reeks te kopiëren, moet u een tweedimensionele reeks (een reeks met rijen en kolommen) opgeven met DECLARE.

Tenzij u een bereik opgeeft, begint het proces met het eerste record in het database-bestand en wordt het proces voortgezet totdat er geen records in het database-bestand of geen rijen in de reeks meer zijn.

De gegevenstypen van de reekselementen zullen hetzelfde zijn als de overeenkomstige veldtypen in het database-bestand.

Opties

Dit commando probeert alle velden (behalve de memovelden) te kopiëren, tenzij u de optie FIELDS of het commando SET FIELDS gebruikt.

Als u de FOR-clausule gebruikt, wordt de voorwaarde eerst geëvalueerd voordat elk record in het database-bestand naar de reeks wordt gekopieerd. Een record wordt alleen naar de reeks gekopieerd als het resultaat van de voorwaarde logisch waar (.T.) is. Als u de WHILE-clausule gebruikt, worden er geen gegevens meer gekopieerd wanneer het resultaat van de voorwaarde logisch onwaar (.F.) is.

Voorbeeld

In het volgende voorbeeld worden de records in het database-bestand Transact, waarvan het Klant_nr L0001 is, met COPY gekopieerd. Hiervoor wordt eerst bepaald hoeveel elementen voor de reeks nodig zijn. Daarna wordt een reeks met de naam Records gedefinieerd en vervolgens wordt het commando COPY TO ARRAY uitgevoerd.

```
. USE Transact                    && Transact.dbf bevat vijf velden.
. DECLARE [RECCOUNT(),5]          && Definieer de reeks.
. COPY TO ARRAY Records FOR Klant_nr = "L00001"
      2 records gekopieerd
```

Als RECCOUNT() meer dan 234 records oplevert, wordt de foutmelding **Onjuiste array-afmetingen** weergegeven. De reden hiervoor is dat het maximum van de array-afmetingen 1170 is (5 x 234).

Zie ook

APPEND FROM ARRAY, COPY FILE, COPY STRUCTURE, DECLARE, EXPORT, SET DELETED, SET FIELDS, SET SAFETY

COUNT

Met COUNT wordt bijgehouden hoeveel records in het actieve database-bestand met de opgegeven voorwaarde overeenkomen.

Syntaxis

COUNT [TO <geheugenvariabele>] [<bereik>] [FOR <voorwaarde>]
[WHILE <voorwaarde>]

Gebruik

Als SET TALK op ON is ingesteld, wordt met dit commando het aantal records geteld en wordt de score weergegeven. Als u met een FOR- of WHILE-clausule een voorwaarde opgeeft of het aantal records met een bereik begrenst, is de score een aanduiding van het aantal records dat aan de voorwaarde voldoet of binnen het bereik valt. COUNT heeft alleen invloed op velden die in het commando SET FILTER TO zijn opgenomen.

Als u de TO-clausule opneemt, wordt met COUNT zonodig een geheugenvariabele gemaakt waarin de score als een getal van het type N (binair-gecodeerd decimaal getal) wordt opgeslagen.

Speciale gevallen

In een netwerkomgeving wordt het bestand tijdens de verwerking door COUNT automatisch vergrendeld als SET LOCK op ON is ingesteld (de standaardinstelling) en na het tellen weer ontgrendeld. Als SET LOCK op OFF is ingesteld, kunt u met COUNT ook een telling uitvoeren maar het resultaat kan onbetrouwbaar zijn als een andere gebruiker het database-bestand ondertussen wijzigt.

Voorbeeld

In het volgende voorbeeld wordt het aantal orders in de maand maart in het database-bestand Transact bepaald:

```
. USE Transact
. COUNT FOR LIKE("??-03-87",DTOC(Dat_trans)) TO Maart_Tel
                    6 records
. ? "Er waren in maart "+LTRIM(STR(Maart_Tel,3,0))+" orders."
Er waren in maart 6 orders.
```

Zie ook

AVERAGE, CALCULATE, RECCOUNT(), SUM

CREATE FROM

Met CREATE FROM wordt een nieuw database-bestand gevormd uit de structuur die met het commando COPY STRUCTURE EXTENDED is gemaakt.

Syntaxis

CREATE <bestandsnaam> FROM <structuur uitgebreid bestand>

Gebruik

Dit commando wordt het meest in toepassingsprogramma's in combinatie met het commando COPY STRUCTURE EXTENDED gebruikt. Met CREATE FROM wordt uit het uitgebreide structuurbestand een database-bestand gemaakt zonder gebruik te maken van de commando's CREATE/MODIFY STRUCTURE.

Het bestand dat met CREATE FROM is gemaakt, wordt het actieve database-bestand in het actieve werkgebied. Als de CREATE FROM-bewerking om een bepaalde reden niet lukt, is er geen geopend database-bestand in het actieve werkgebied.

Tip

Gebruik als naam van het database-bestand niet de letters A tot en met J of de letter M aangezien deze letters zijn gereserveerd voor de standaard-aliasnamen. U kunt echter wel bijvoorbeeld AA als naam van het database-bestand opgeven.

Als er zich in het bestand dat met COPY STRUCTURE EXTENDED is gemaakt, velden met FIELD_IDX = J bevinden, wordt er een bijbehorend .mdx-bestand gemaakt en worden de bijbehorende labels in het nieuwe bestand geplaatst.

Voorbeeld

In het volgende voorbeeld wordt met het commando COPY STRUCTURE EXTENDED het nieuwe database-bestand Plusnaam uit Afnemers.dbf gemaakt. Daarna wordt met CREATE FROM een nieuw database-bestand met dezelfde structuur als Afnemers.dbf gemaakt.

```
. USE Afnemers
. COPY STRUCTURE EXTENDED TO Plusnaam
. CREATE Nafnemrs FROM Plusnaam
. DISPLAY STRUCTURE
Structuur van database:    C:\DBASE\NAFNEMRS.DBF
Aantal records:            0
Laatst gewijzigd op:       11-05-87
    Veld        Veldnaam        Type        Breedte     Dec     Index
       1        KLANT_NR        Teken         6                   J
       2        KLANTNAAM       Teken        30                   J
       3        ACHTERNAAM      Teken        15                   N
       4        VOORNAAM        Teken        15                   N
       5        ADRES           Teken        30                   N
       6        WOONPLAATS      Teken        20                   N
       7        PROVINCIE       Teken         2                   N
       8        POSTCODE        Teken        10                   N
       9        TELEFOON        Teken        13                   N
      10        OVERZICHT       Memo         10                   N
** Totaal **                               152
```

Zie ook

COPY, COPY STRUCTURE, COPY STRUCTURE EXTENDED, LIST/DISPLAY STRUCTURE

CREATE/MODIFY APPLICATION

Met CREATE/MODIFY APPLICATION hebt u toegang tot de applicatiegenerator van dBASE IV. Hiermee wordt de code gegenereerd die objecten, zoals database-bestanden, indexbestanden, queries, rapporten, formulieren, menu's en lijsten, in één toepassing samenbrengt.

Syntaxis

CREATE/MODIFY APPLICATION <bestandsnaam>/?

Standaardinstellingen

Voor de toepassing geeft u in de commandoregel een bestandsnaam op of gebruikt u de vraagclausule (?). Als u een bestandsnaam voor de toepassing opgeeft en een nieuwe toepassing maakt, gebruikt dBASE IV deze naam in het dialoogvenster **Definitie toepassing**. U kunt de naam van de toepassing echter in het dialoogvenster wijzigen.

Gebruik

De commando's CREATE APPLICATION en MODIFY APPLICATION zijn identiek. De aanwezigheid van een toepassingsobjectbestand (.app) en niet het gebruikte commando bepaalt of er een bestand wordt gemaakt of wordt gewijzigd. Als het .app-objectbestand bestaat, kunt u het met dit commando wijzigen en als het .app-objectbestand niet bestaat, kunt u het met dit commando maken.

Met de applicatiegenerator kunt u behalve toepassingsobjecten, de volgende objecten maken:

Tabel 2-5 Objecten die met de applicatiegenerator kunnen worden gemaakt

Object	Extensie	Omschrijving
Horizontaal balkmenu	.bar	Menu-opties die van links naar rechts op het scherm worden weergegeven
Pop-up menu	.pop	Menu-opties die verticaal in een kader op het scherm worden weergegeven
Bestandenlijst	.fil	Een lijst met bestanden waaruit u kunt kiezen
Structuurlijst	.str	Een lijst van velden uit het actieve database-bestand of de actieve visie waaruit u kunt kiezen
Waardenlijst	.val	Een lijst met waarden die een veld kan bevatten
Batch-proces	.bch	Een serie handelingen die hoort bij een menu-optie of een lijst en die door de toepassing kan worden uitgevoerd

Als er een catalogus is geopend, wordt alleen het toepassingsobject (.app) dat u maakt, aan de catalogus toegevoegd. De andere objecten die u maakt (.bar, .pop, .fil, .str, .val of .bch), worden aan de actieve sub-bestandsindex toegevoegd.

Opties

Als er een catalogus is geopend, wordt in de catalogus met het ?-symbool naar alle beschikbare .app-objectbestanden gezocht. Als er geen catalogus is geopend, worden met het ?-symbool alle opgeslagen .app-objectbestanden getoond. U kunt dan de toepassing die u wilt wijzigen, kiezen.

Speciale gevallen

Vanuit het Control Center hebt u ook toegang tot de applicatiegenerator. Als u de applicatiegenerator echter via CREATE/MODIFY APPLICATION oproept en u deze verlaat, wordt de besturing aan de commandostip of aan het volgende commando in een programmabestand overgedragen en niet aan het Control Center.

Als u uit het paneel **Toepassing** van het Control Center de markering **<nieuw>** kiest, kunt u beslissen of u met de tekstverwerker of met de applicatiegenerator een nieuwe toepassing maakt. Met CREATE/MODIFY APPLICATION wordt de applicatiegenerator echter direct op het scherm weergegeven. De tekstverwerker roept u op door MODIFY COMMAND te typen.

Zie ook

Werken met de dBASE IV applicatiegenerator bevat verdere informatie over toepassingen en hoe u deze maakt.

CREATE/MODIFY LABEL

Met CREATE/MODIFY LABEL hebt u toegang tot het etiketontwerpscherm. Met het etiketontwerpscherm kunt u etiketformulierbestanden (.lbl) maken met behulp van de velden die in het actieve database-bestand of in andere gerelateerde database-bestanden zijn opgegeven.

Syntaxis

CREATE/MODIFY LABEL <bestandsnaam>/?

Standaardinstellingen

Tenzij u een andere extensie opgeeft, geeft dBASE IV de naam van het bestand de extensie .lbl en genereert een bestand waarvan de naam de extensie .lbg heeft. Als er een catalogus is geopend en SET CATALOG op ON is ingesteld, voegt dBASE IV het etiketbestand aan de catalogus toe.

Gebruik

Met het commando CREATE LABEL maakt u een nieuw etiketformulierbestand (.lbl). Het .lbl-bestand bevat alle gegevens die nodig zijn om het etiketontwerpscherm dat later weer kan worden gewijzigd, te kunnen weergeven en om op basis van de gegevens uit een database-bestand of visie, etiketten te kunnen afdrukken.

Van het etiket dat u wilt maken, geeft u de grootte op en ook het aantal af te drukken regels op elk etiket. Deze waarden moeten overeenkomen met de etiketformulieren of het papier waarop u wilt afdrukken.

De commando's CREATE LABEL en MODIFY LABEL zijn identiek. De aanwezigheid van een etiketformulierbestand en niet het gebruikte commando bepaalt of er een bestand wordt gemaakt of wordt gewijzigd. Als het .lbl-bestand bestaat, kunt u het met dit commando wijzigen en als het .lbl-bestand niet bestaat, kunt u het met dit commando maken.

Wanneer u CREATE/MODIFY LABEL gebruikt, selecteert u de gegevens die u op de etiketten wilt laten afdrukken en maakt u deze op. Als u het etiketformulier opslaat, maakt CREATE/MODIFY LABEL een bestand met dBASE IV-code voor het afdrukken van etiketten. Dit bestand heeft dezelfde naam als het etiketformulierbestand maar heeft de extensie .lbg. Wanneer u de etiketten voor de eerste maal met het commando LABEL FORM afdrukt, wordt een .lbo-bestand dat de gecompileerde objectcode van het .lbg-bestand bevat, opgeslagen. Telkens wanneer u daarna etiketten met dit formulier afdrukt, wordt door het commando LABEL FORM dit .lbo-bestand gebruikt.

Opties

Als er een catalogus is geopend, wordt in de catalogus met het ?-symbool naar alle beschikbare .lbl-bestanden gezocht die horen bij het actieve database-bestand of de actieve visie. Als er geen catalogus is geopend, worden met het ?-symbool alle opgeslagen .lbl-bestanden getoond. U kunt dan het etiketformulierbestand kiezen dat u wilt wijzigen.

Speciale gevallen

Het etiketontwerpscherm kunt u ook vanuit het Control Center oproepen. Als u het etiketontwerpscherm echter met CREATE/MODIFY LABEL oproept en u dit verlaat, wordt de besturing aan de commandostip of aan het volgende commando in een programmabestand overgedragen en niet aan het Control Center. Als SET DESIGN is ingeschakeld, kunt u het query-ontwerpscherm uit BROWSE of EDIT afbeelden door op **SHIFT-F2 Design** te drukken.

Als u het etiketformulierbestand (.lbl) wist, kunt u het bestand niet met de commando's CREATE of MODIFY LABEL bewerken. Wanneer u het bestand probeert te bewerken, wordt er een nieuw etiketbestand gegenereerd als het oude etiketbestand niet wordt aangetroffen.

Hoewel u met MODIFY COMMAND of met een tekstverwerker rechtstreeks wijzigingen aan het gegenereerde etiketbestand (.lbg) kunt aanbrengen, blijven de .lbl- of .lbo-bestanden daarna ongewijzigd. Met LABEL FORM kunt u een nieuw .lbo-bestand compileren. Door LABEL FORM wordt de tijd vergeleken waarop de .lbo- en de .lbg-bestanden zijn gemaakt. Aan de hand hiervan wordt bepaald wanneer er een nieuw .lbo-bestand moet worden gecompileerd.

Als u een etiketformulierbestand (.lbl) dat in dBASE III PLUS is gemaakt, wijzigt, wordt het bestand naar de indeling van dBASE IV geconverteerd en krijgt de naam van het oorspronkelijke .lbl-bestand van dBASE III PLUS de extensie .lb3. In dBASE III PLUS kunt u geen dBASE IV-etiketten gebruiken. Etiketformulieren die met dBASE III PLUS zijn gemaakt, kunnen zonder wijzigingen in dBASE IV worden gebruikt.

Zie ook

DO, LABEL FORM, SET CATALOG, SET SAFETY, SET VIEW, _pageno

Menu's van dBASE IV bevat verdere informatie over het maken van etiketten met het etiketontwerpscherm van dBASE IV.

CREATE/MODIFY QUERY/VIEW

Met CREATE/MODIFY QUERY of CREATE/MODIFY VIEW hebt u toegang tot het query-ontwerpscherm. Met dit scherm kunt u query-bestanden (.qbe) maken waarmee records worden opgehaald die aan opgegeven voorwaarden voldoen of mutatie-query-bestanden (.upd) maken waarmee records in het database-bestand worden gewijzigd. Met een query-bestand kunt u indexen activeren, sorteerbewerkingen uitvoeren en totalen berekenen.

Syntaxis

CREATE/MODIFY QUERY <bestandsnaam>/?

of

CREATE/MODIFY VIEW <bestandsnaam>/?

Standaardinstellingen

Tenzij u een andere extensie opgeeft, geeft dBASE IV de naam van het query-bestand bij het wegschrijven de extensie .qbe of .upd. Als er een catalogus is geopend en SET CATALOG op ON is ingesteld, wordt het query-bestand aan de catalogus toegevoegd.

Gebruik

Door een .qbe-bestand worden alleen records weergegeven die aan de opgegeven voorwaarden voldoen, wanneer daarna commando's volgen. Een .upd-bestand bevat opdrachten voor het bijwerken van records in het database-bestand.

De naam van het nieuwe query-bestand krijgt alleen de extensie .upd als u in de bestandsstructuur van het ontwerpscherm onder de bestandsnaam een mutatie-operator plaatst.

Met zowel CREATE/MODIFY QUERY als CREATE/MODIFY VIEW hebt u toegang tot het query-ontwerpscherm.

De vormen CREATE en MODIFY van het commando zijn identiek. De aanwezigheid van een query-bestand (.qbe of .upd) en niet het gebruikte commando bepaalt of er een bestand wordt gemaakt of wordt gewijzigd. Als het .qbe- of .upd-bestand bestaat, wordt op het scherm de vraag weergegeven of u het wilt wijzigen en als het .qbe- of .upd-bestand niet bestaat, wordt het met dit commando gemaakt.

dBASE IV-visies zijn een omvattende verzameling van de queries en visies die in eerdere versies van dBASE zijn gemaakt. Dit commando kan de .vue-bestanden die met eerdere versies zijn gemaakt, lezen en daaruit nieuwe .qbe-bestanden maken. De nieuwe .qbe-bestanden kunnen echter niet door eerdere versies van dBASE worden gelezen.

Als u geen extensie opgeeft, wordt met dit commando eerst naar een .qbe- of .upd-bestand gezocht dat eerder met het query-ontwerpscherm is gemaakt. Wanneer er geen .qbe- of .upd-bestand wordt aangetroffen, wordt met dit commando naar een .vue-bestand gezocht dat met een eerdere versie van dBASE is gemaakt. Als er ook geen .vue-bestand wordt aangetroffen, wordt door dit commando een nieuw bestand gemaakt waarvan de naam de extensie .qbe of .upd krijgt.

CREATE/MODIFY QUERY/VIEW

Wanneer u een .qbe-bestand voor de eerste maal met SET VIEW activeert, wordt er een .qbo-bestand opgeslagen. Steeds wanneer u daarna deze query activeert, wordt door het SET VIEW-commando dit .qbo-bestand gebruikt.

Wanneer u de mutatie-query met DO uitvoert, wordt er een .dbo-bestand opgeslagen. Deze .dbo-extensie kunt u wijzigen in .upo als u de mutatie-query-bestanden van de programmabestanden wilt scheiden (de extensie moet u echter in .upo wijzigen als u dit

ASSIST

Met ASSIST krijgt u toegang tot het Control Center van dBASE IV.

Syntaxis

ASSIST

Gebruik

Met ASSIST wordt het Control Center weergegeven, van waaruit u de verschillende delen van het menusysteem van dBASE IV kunt oproepen.

Het Control Center opent altijd een catalogus. Eerst wordt geprobeerd de laatst geopende catalogus in de hoofdcatalogus te openen. Als geen catalogus beschikbaar is, wordt een nieuwe catalogus gemaakt met de naam Naamloos.cat.

Om het menusysteem te gebruiken, volgt u de instructies op het scherm op.

Het Control Center geeft u toegang tot zes ontwerpschermen die worden weergegeven als panelen in het Control Center. Deze ontwerpschermen kunnen ook worden opgeroepen met de commando's CREATE, CREATE/MODIFY QUERY/VIEW, CREATE/MODIFY SCREEN, CREATE/MODIFY REPORT, CREATE/MODIFY LABEL en CREATE/MODIFY APPLICATION.

U kunt daarnaast gegevens bekijken, bewerken en toevoegen, zoals u dit doet met BROWSE, EDIT en APPEND. U kunt rapporten en etiketten maken, zoals met REPORT FORM en LABEL FORM en u kunt programma's en andere bestanden uitvoeren, zoals met DO, SET VIEW en SET FORMAT. Bovendien kunt u de tekstverwerker gebruiken, zoals met MODIFY COMMAND/FILE.

Om het Control Center te verlaten en terug te keren naar de commandostip, drukt u op **Esc**. U kunt ook op **Alt-S** drukken om het menu **Stop** te openen en de optie **Terug naar commandostip** kiezen.

Zie ook

Het Control Center wordt beschreven in *dBASE IV voor beginners* en *Menu's van dBASE IV*. Raadpleeg deze boeken voor meer informatie.

CREATE/MODIFY REPORT

Met CREATE/MODIFY REPORT hebt u toegang tot het rapportontwerpscherm. Met dit scherm kunt u rapportformulierbestanden (.frm) maken op basis van de velden die in het actieve database-bestand of in andere gerelateerde database-bestanden zijn opgegeven.

Syntaxis

CREATE/MODIFY REPORT <bestandsnaam>/?

Standaardinstellingen

Tenzij u een andere extensie opgeeft, geeft dBASE IV de naam van het bestand de extensie .frm. Met CREATE REPORT wordt altijd een bestand gemaakt waarvan de naam de extensie .frg heeft. Als er een catalogus is geopend en SET CATALOG op ON is ingesteld, wordt het rapportformulierbestand aan de catalogus toegevoegd.

Gebruik

Het commando CREATE/MODIFY REPORT wordt gebruikt voor het maken van een nieuw rapportformulierbestand (.frm). Het rapportformulierbestand bevat alle gegevens die nodig zijn voor het instellen van de weergave van het rapportontwerp (dat kan worden gewijzigd) en voor het afdrukken van rapporten met gegevens uit een database-bestand of visie.

De commando's CREATE REPORT en MODIFY REPORT zijn identiek. De aanwezigheid van een rapportformulierbestand (.frm) en niet het gebruikte commando bepaalt of er een bestand wordt gemaakt of wordt gewijzigd. Als het .frm-bestand bestaat, wordt het met dit commando gewijzigd en als het .frm-bestand niet bestaat, wordt het met dit commando gemaakt.

Met CREATE of MODIFY REPORT kunt u de gegevens selecteren die het rapport moet bevatten, de velden plaatsen waar deze moeten worden afgedrukt, gegevens in groepen samenbrengen en statistische gegevens met numerieke uitdrukkingen berekenen. Wanneer u het rapportformulier opslaat, maakt CREATE/MODIFY REPORT een bestand met de dBASE IV-code voor het afdrukken van rapporten. Dit bestand heeft dezelfde naam als het rapportformulierbestand, maar met de extensie .frg. Wanneer u het rapport afdrukt, wordt, net als met het commando REPORT FORM, een .fro-bestand met de gecompileerde objectcode van het .frg-bestand opgeslagen. Steeds wanneer u daarna een rapport met dit formulier afdrukt, gebruikt het commando REPORT FORM dit .fro-bestand.

Opties

Als er een catalogus is geopend, wordt in de catalogus met het ?-symbool naar alle beschikbare .frm-bestanden gezocht die horen bij het actieve database-bestand of de actieve visie. Als er geen catalogus is geopend, worden met het ?-symbool alle opgeslagen .frm-bestanden getoond. U kunt dan het rapportformulierbestand kiezen dat u wilt wijzigen.

Speciale gevallen

U kunt het rapportontwerpscherm ook vanuit het Control Center oproepen. Als u echter het rapportontwerpscherm met CREATE/MODIFY REPORT oproept en u dit verlaat, wordt de besturing aan de commandostip of aan het volgende commando in een programmabestand overgedragen en niet aan het Control Center.

Als u het rapportformulierbestand (.frm) wist, kunt u dit bestand niet met de commando's CREATE of MODIFY REPORT bewerken. Wanneer u dit bestand probeert te bewerken, wordt er een nieuw rapportbestand gegenereerd als het oude rapportbestand niet wordt aangetroffen.

Hoewel u met MODIFY COMMAND of met een tekstverwerker rechtstreeks wijzigingen aan het gegenereerde rapportformulierbestand (.frg) kunt aanbrengen, blijven de .frm- of .fro-bestanden daarna ongewijzigd. Met REPORT FORM compileert u een nieuw .fro-bestand.

Als u een rapportformulierbestand (.frm) dat in dBASE III PLUS is gemaakt, wijzigt, wordt het bestand naar de indeling van dBASE IV geconverteerd en krijgt de naam van het oorspronkelijke .frm-bestand van dBASE III PLUS de extensie .fr3. In dBASE III PLUS kunt u geen dBASE IV-rapporten gebruiken.

Zie ook

REPORT FORM, SET CATALOG, SET DESIGN, SET SAFETY, SET VIEW, _pageno

Menu's van dBASE IV bevat verdere informatie over het maken van rapporten met het rapportontwerpscherm.

CREATE/MODIFY SCREEN

Met CREATE/MODIFY SCREEN hebt u toegang tot het formulierontwerpscherm waarmee u aangepaste schermformulieren kunt maken. Met deze schermformulieren bepaalt u de manier waarop velden en andere gegevens op het scherm worden weergegeven wanneer u een schermgroot bewerkcommando zoals EDIT of APPEND gebruikt.

Syntaxis

CREATE/MODIFY SCREEN <bestandsnaam>/?

Standaardinstellingen

Tenzij u andere extensies opgeeft, geeft dBASE IV de naam van het schermbestand bij het wegschrijven de extensie .scr en genereert een indelingsbestand waarvan de naam de extensie .fmt heeft. Als er een catalogus is geopend en SET CATALOG op ON is ingesteld, worden de query-bestanden en indelingsbestanden aan de catalogus toegevoegd.

Wanneer u een nieuw indelingsbestand maakt, wordt dit automatisch geopend. Zodra u daarna een commando invoert waarvoor een schermbestand nodig is (APPEND, CHANGE, EDIT, INSERT of READ), wordt de nieuwe indeling gebruikt.

Gebruik

Met het commando CREATE SCREEN wordt een schermbestand (.scr) en een nieuw indelingsbestand (.fmt) gemaakt. Het schermbestand bevat de gegevens in een vorm die u kunt wijzigen en het indelingsbestand bevat de dBASE IV-commando's waarmee de gegevens op het scherm worden weergegeven. Voor het ontwerpen of wijzigen van een scherm werkt u met het schermbestand.

De commando's CREATE SCREEN en MODIFY SCREEN zijn identiek. De aanwezigheid van een schermbestand (.scr) en niet het gebruikte commando bepaalt of er een bestand wordt gemaakt of wordt gewijzigd. Als het .scr-bestand bestaat, wordt het met dit commando gewijzigd en als het .scr-bestand niet bestaat, wordt het met dit commando gemaakt en wordt er een nieuw indelingsbestand gegenereerd.

Met CREATE of MODIFY SCREEN kunt u de positie van de velden op het scherm bepalen, berekende velden en geheugenvariabelen weergeven en extra tekst, kaders en lijnen opnemen, en velden een bepaalde kleur meegeven, kaders en blokken tekst selecteren, verplaatsen en kopiëren.

Wanneer u het schermformulier opslaat, wordt door CREATE/MODIFY SCREEN een indelingsbestand met @-commando's gemaakt waarmee de gegevens kunnen worden weergegeven en bewerkt. Het indelingsbestand heeft dezelfde naam als het schermbestand, maar met de extensie .fmt. Wanneer u met het commando SET FORMAT het indelingsbestand voor de eerste maal gebruikt, wordt een .fmo-bestand met de gecompileerde objectcode van het .fmt-bestand opgeslagen. Steeds wanneer u daarna SET FORMAT met een schermgroot bewerkcommando gebruikt, wordt dit .fmo-bestand door dBASE IV gebruikt.

Wanneer u een indelingsbestand wilt gebruiken, moet u het bijbehorende .dbf-bestand met USE openen. Daarna opent u het indelingsbestand met SET FORMAT TO <naam indelingsbestand> (tenzij u dit bestand zojuist hebt gemaakt). Als u geen indelingsbestand opent, wordt door APPEND, BROWSE, CHANGE, EDIT, INSERT en READ een standaard-invoerscherm gebruikt. Dit scherm lijkt op het scherm Basisopmaak.

Het indelingsbestand dat u hebt gemaakt, kunt u op elk gewenst moment openen en sluiten. Met SET FORMAT TO <naam indelingsbestand> kunt u een indelingsbestand openen en met SET FORMAT TO (zonder parameter) sluit u het weer. U kunt een indelingsbestand ook met CLOSE FORMAT sluiten.

Opties

Als er een catalogus is geopend, wordt in de catalogus met het ?-symbool naar alle beschikbare .scr-bestanden gezocht die horen bij het actieve database-bestand of de actieve visie. Als er geen catalogus is geopend, worden met het ?-symbool alle opgeslagen .scr-bestanden getoond. U kunt dan het schermbestand kiezen dat u wilt wijzigen.

Speciale gevallen

In tegenstelling tot eerdere versies van dBASE blijft de fysieke structuur van het database-bestand met CREATE/MODIFY SCREEN ongewijzigd.

Als u het schermbestand wist, kunt u dit bestand niet met de commando's CREATE of MODIFY SCREEN bewerken. Wanneer u dit bestand probeert te bewerken, wordt er een nieuw schermbestand gegenereerd als het oude schermbestand niet wordt aangetroffen.

Hoewel u met MODIFY COMMAND of met een tekstverwerker rechtstreeks wijzigingen aan het indelingsbestand (.fmt) kunt aanbrengen, blijven de .scr- of .fmo-bestanden daarna ongewijzigd. Met SET FORMAT compileert u een nieuw .fmo-bestand.

Als u een schermbestand (.scr) dat in dBASE III PLUS is gemaakt, wijzigt, wordt het bestand naar de indeling van dBASE IV geconverteerd en krijgt de naam van het oorspronkelijke .scr-bestand van dBASE III PLUS de extensie .sc3. In dBASE III PLUS kunt u geen dBASE IV-schermbestanden wijzigen.

Het indelingsbestand dat door CREATE/MODIFY SCREEN wordt gegenereerd, bevat letterlijke tekenreeksen tussen dubbele aanhalingstekens. Vanwege de dubbele aanhalingstekens in het indelingsbestand, kunt u deze beter niet gebruiken in het bovenliggende ontwerp van een schermformulier.

Zie ook

@, APPEND, BROWSE, EDIT, INSERT, READ, SET CATALOG, SET FORMAT, SET SAFETY, SET VIEW

Menu's van dBASE IV bevat verdere informatie over het gebruik van het ontwerpscherm.

CREATE/MODIFY STRUCTURE

Met CREATE/MODIFY STRUCTURE hebt u toegang tot het ontwerpscherm van het database-bestand.

Met CREATE STRUCTURE kunt u een structuur voor een nieuw database-bestand samenstellen. Met MODIFY STRUCTURE kunt u de structuur van een bestaand database-bestand wijzigen.

De structuur van een database-bestand bestaat uit de definitie van de veldnamen, de veldtypen, de veldlengten, het aantal decimale posities (voor numerieke velden) en een vlag die voor elk veld de aanwezigheid van een .mdx-label aanduidt.

Syntaxis

CREATE <bestandsnaam>/MODIFY STRUCTURE

Standaardinstellingen

Tenzij u bij de bestandsnaam een station en een bestandsindex opgeeft, wordt door CREATE het nieuwe database-bestand naar het standaardstation en de standaardbestandsindex weggeschreven. Tenzij u een andere extensie opgeeft, krijgt de naam van het database-bestand de extensie .dbf.

Bij MODIFY STRUCTURE geeft u geen bestandsnaam op. Dit commando kan alleen de structuur van het actieve database-bestand wijzigen.

Als er een catalogus actief is wanneer u een database-bestand maakt, wordt het bestand aan deze catalogus toegevoegd.

Gebruik

Met CREATE <nieuw bestand> en MODIFY STRUCTURE hebt u hetzelfde ontwerpscherm voor het maken en wijzigen van de structuur van het database-bestand. De structuur bevat definities voor elk veld in het database-bestand.

U kunt voor <bestandsnaam> een indirecte verwijzing gebruiken. Een indirecte verwijzing is een tekenuitdrukking die resulteert in een bestandsnaam en die u overal kunt gebruiken waar u een bestandsnaam moet opgeven. In deze uitdrukking moet u een operator gebruiken (meestal haakjes), zodat dBASE weet dat de tekenreeks een uitdrukking en geen letterlijke bestandsnaam is. Een indirecte verwijzing gebruikt u op dezelfde manier als het macrosubstitutie-teken, maar de indirecte verwijzing wordt veel sneller uitgevoerd.

CREATE/MODIFY STRUCTURE

De structuur van een nieuw database-bestand definieert u door voor elk veld de volgende gegevens op te geven:

veldnaam
type
breedte
decimale posities (voor een numeriek veld)
veldindexvlag (als het resultaat waar (.T.) is, wordt aan het bijbehorende .mdx-bestand een label toegevoegd met een index op dit veld)

De veldnaam mag maximaal 10 tekens bevatten en kan bestaan uit letters, cijfers en het onderstrepingsteken. De veldnaam mag geen ingebedde spaties bevatten en het eerste teken van de veldnaam moet een letter zijn. Wanneer u de veldnaam hebt ingevoerd, drukt u op ↵.

Het veldtype bepaalt u door de eerste letter van het gegevenstype in te voeren (**T**eken [tekentype], **N**umeriek [binair-gecodeerd decimaal numeriek getal], Zwevende komma [numeriek getal met zwevend decimaalteken], **L**ogisch [logisch type], **D**atum [datumtype] of **M**emo [memotype]) of door op de **spatiebalk** te drukken totdat het gewenste gegevenstype wordt weergegeven en dit vervolgens te selecteren door op ↵ te drukken.

Voor numerieke velden en tekenvelden moet u een veldbreedte opgeven. Deze breedte is het maximum aantal cijfers of tekens dat u in het veld wilt invoeren. Tekenvelden mogen maximaal 254 tekens lang zijn en numerieke velden maximaal 20 cijfers, inclusief het (valuta)teken en een decimaalteken.

Logische velden, datumvelden en memovelden hebben een vooraf gedefinieerde breedte. Een logisch veld is één byte breed. Datumvelden zijn altijd 8 bytes. Memovelden wordt in het database-bestand automatisch de lengte van 10 bytes toegewezen hoewel elke invoer in het memoveld tot 64 Kbyte aan gegevens kan bevatten. De gegevens die u in memovelden invoert, worden niet in het database-bestand (.dbf) opgeslagen maar in het bijbehorende memotekstbestand (.dbt). Met de 10 bytes wordt de lokatie van de invoer in het memoveld aangeduid.

Als u in de kolom Index van het ontwerpscherm een J opgeeft, maakt dBASE IV in het bijbehorende .mdx-bestand een indexlabel op dat veld.

Voor een record kunt u maximaal 255 velden definiëren. De maximum grootte van een record is 4000 bytes, inclusief 10 bytes voor elk memoveld. Elke tekenpositie in een veld neemt één byte in beslag.

Opdrachten en foutmeldingen worden onderaan het scherm weergegeven. Door de afrolmenu's bovenaan het scherm kunt u rechtstreeks met de structuur en de records van het database-bestand werken. U kunt de structuur van het database-bestand afdrukken, indexen maken, het bestand sorteren, indexen verwijderen en records aan het bestand toevoegen.

CREATE/MODIFY STRUCTURE

Tips

Met MODIFY STRUCTURE wordt van het database-bestand een reservekopie met de extensie .bak en van het memobestand met de extensie .tbk gemaakt in dezelfde bestandsindex als het oorspronkelijke bestand. Nadat de wijzigingen in de structuur zijn aangebracht, wordt de inhoud uit de reservekopiebestanden aan het gewijzigde database-bestand toegevoegd. Aangezien MODIFY STRUCTURE geen reservekopiebestanden kan maken wanneer de diskette of de actieve bestandsindex vol is, moet u ervoor zorgen dat er voldoende ruimte voor de reservekopiebestanden beschikbaar is voordat u de structuur van een bestand wijzigt.

WAARSCHUWING
Het .bak-bestand dat met MODIFY STRUCTURE wordt gemaakt, wordt door een volgend MODIFY STRUCTURE-commando overschreven.

Aangezien door MODIFY STRUCTURE gegevens aan het nieuwe bestand worden toegevoegd, kan er gegevensverlies optreden wanneer u tijdens het opslaan van de wijzigingen de computer onderbreekt.

Speciale gevallen

In een netwerkomgeving moet u het exclusieve gebruik van het database-bestand hebben voordat u de structuur daarvan kunt wijzigen.

Het verdient aanbeveling de veldnaam en de veldbreedte of het gegevenstype niet tegelijkertijd te wijzigen. Als u dit namelijk doet, kan dBASE IV de gegevens uit het oude veld niet aan het nieuwe veld toevoegen en blijft dit leeg. Wijzig eerst de veldnaam, sla het bestand op en gebruik daarna opnieuw MODIFY STRUCTURE om de veldbreedte of het gegevenstype te wijzigen.

Het verdient bovendien aanbeveling niet tegelijkertijd velden uit een database-bestand in te voegen of te wissen en veldnamen te wijzigen. Als u veldnamen wijzigt, voegt MODIFY STRUCTURE op basis van de veldpositie in het bestand gegevens uit het oude bestand toe. Als u daarom velden invoegt of wist en tevens veldnamen wijzigt, worden de veldposities gewijzigd en kan er gegevensverlies optreden. U kunt echter wel tegelijkertijd veldbreedten of gegevenstypen wijzigen en velden invoegen of wissen. In dergelijke gevallen worden de gegevens correct toegevoegd omdat MODIFY STRUCTURE de gegevens op basis van de veldnaam toevoegt.

dBASE IV kan voor een aantal veldtypeconversies de gegevens op de juiste manier converteren. Als u echter het veldtype wijzigt, maakt u eerst een reservekopie van het oorspronkelijke bestand zodat u kunt controleren of de gegevens correct naar de nieuwe bestanden zijn geconverteerd.

Als u numerieke velden naar tekenvelden converteert, worden de getallen uit de numerieke velden door dBASE IV naar rechtsgelijnde tekenreeksen geconverteerd. Als u een tekenveld naar een numeriek veld converteert, worden de numerieke tekens in de records door dBASE IV naar cijfers geconverteerd totdat een niet-numeriek teken wordt aangetroffen. Als het eerste teken in een tekenveld een letter is, bevat het geconverteerde numerieke veld nul tekens.

U kunt logische velden naar tekenvelden converteren of omgekeerd. U kunt logische velden niet naar numerieke velden converteren. U kunt ook tekenreeksen die als datum (bijvoorbeeld dd-mm-jj of dd/mm/jj) zijn ingedeeld, naar een datumveld converteren of datumvelden converteren naar tekenvelden.

Als u de veldnaam, de veldlengte of het gegevenstype van een veld wijzigt dat in het bijbehorende .mdx-bestand een label heeft, wordt de label opnieuw samengesteld.

dBASE IV probeert met betrekking tot de conversie van het gegevenstype van velden in het algemeen de gewenste conversie uit te voeren. Wanneer de conversie echter niet goed is overwogen, kan er gegevensverlies optreden. Numerieke gegevens kunnen gemakkelijk als tekens worden verwerkt, maar logische gegevens bijvoorbeeld kunnen geen numerieke gegevens worden.

Als u incompatibele gegevenstypen wilt converteren, bijvoorbeeld logisch naar numeriek, voegt u eerst een nieuw veld aan het bestand toe, converteert u de gegevens met REPLACE en verwijdert u ten slotte het oude veld.

Als u een database-bestand als een SQL-tabel in SQL-modus wilt gebruiken, moet u het bestand definiëren met het SQL-commando DBDEFINE. Met het commando DBDEFINE worden de systeemcatalogi die SQL voor de toegang tot het bestand gebruikt, bijgewerkt.

Zie ook

APPEND FROM, APPEND MEMO, COPY STRUCTURE, COPY STRUCTURE EXTENDED, CREATE FROM, SET BLOCKSIZE, SET EXCLUSIVE, SET SAFETY, SET SQL

CREATE VIEW FROM ENVIRONMENT

Met CREATE VIEW FROM ENVIRONMENT wordt een visiebestand (.vue) samengesteld dat compatibel is met dBASE III PLUS.

Syntaxis

CREATE VIEW <.vue-bestandsnaam>/? FROM ENVIRONMENT

Standaardinstellingen

U moet een bestandsnaam voor het visiebestand opgeven of een ? (query-clausule) op de commandoregel gebruiken. Tenzij u een andere extensie opgeeft, krijgen de namen van de bestanden die u met dit commando maakt de extensie .vue. Als er een catalogus is geopend en SET CATALOG op ON is ingesteld, voegt dBASE IV het visiebestand aan de catalogus toe.

Opties

Als er een geopende catalogus is, wordt met het vraagteken (?) gezocht naar alle .vue-bestanden in de catalogus. Als er geen geopende catalogus is, vertegenwoordigt het vraagteken alle .vue-bestanden op schijf. U kunt vervolgens een bestand kiezen dat u overschrijft.

Gebruik

Met het commando CREATE VIEW FROM ENVIRONMENT van dBASE III PLUS kon u .vue-bestanden maken waarin gegevens over de actieve geselecteerde werkgebieden waren opgeslagen: geopend database-bestand, indelingsbestand en indexbestand, relaties, veldlijsten en filtervoorwaarden. Het commando SET VIEW van dBASE III PLUS activeerde de omgeving die in het .vue-bestand was opgeslagen door de bestanden te openen en de veldlijsten, relaties en filtervoorwaarden opnieuw tot stand te brengen.

Hoewel het query-scherm van dBASE IV, waartoe u met het commando CREATE/MODIFY QUERY/VIEW toegang hebt, veel meer mogelijkheden biedt dan de .vue-bestanden, kunt u met dBASE IV ook .vue-bestanden uit de actieve omgeving maken zodat compatibiliteit met de toepassingen van dBASE III PLUS is gewaarborgd.

Met CREATE VIEW FROM ENVIRONMENT wordt een visiebestand (.vue) samengesteld waarin de volgende gegevens uit de actieve omgeving worden opgeslagen:

- alle geopende database-bestanden, indexbestanden en het werkgebied van elk bestand
- alle relaties tussen de database-bestanden
- het actieve geselecteerde werkgebiednummer
- de actieve veldlijst
- het geopende indelingsbestand (.fmt), indien van toepassing
- de geldende filtervoorwaarden

CREATE VIEW FROM ENVIRONMENT

Voordat u CREATE VIEW FROM ENVIRONMENT gebruikt, moet u de bestanden openen en veldlijsten, filtervoorwaarden en relaties tot stand brengen.

Wanneer u bij SET VIEW TO een .vue-bestand opgeeft, wordt de visie die met CREATE VIEW FROM ENVIRONMENT is opgeslagen, geactiveerd.

U maakt een visiebestand inactief door een ander visiebestand te openen of door CLOSE DATABASES te typen. Hierdoor worden de geopende database-bestanden en de bijbehorende bestanden die het visiebestand vormen, gesloten.

Zie ook

CREATE/MODIFY QUERY/VIEW, SELECT, SET FIELDS, SET FORMAT, SET INDEX, SET RELATION, SET VIEW, USE

DEACTIVATE MENU

Met het commando DEACTIVATE MENU wordt het actieve balkmenu inactief gemaakt en van het scherm verwijderd, maar niet uit het geheugen gewist. Dit commando heeft geen effect, wanneer het vanaf de commandostip wordt uitgevoerd. U gebruikt het in ON SELECTION-opdrachten of in procedures die door ON SELECTION-opdrachten worden aangeroepen.

Syntaxis

DEACTIVATE MENU

Gebruik

Voor dit commando is geen menunaam nodig want alleen het actieve menu wordt inactief gemaakt en van het scherm verwijderd. Wanneer zich onder het inactief gemaakte menu gegevens bevonden, worden die hierna weer weergegeven.

Een inactief gemaakt menu wordt niet uit het geheugen gewist. Met het commando ACTIVATE MENU kunt u het dus op elk moment weer activeren.

DEACTIVATE MENU draagt de besturing over aan de programmaregel die onmiddellijk volgt op de regel waarmee het menu is geactiveerd. Als het commando vanuit een aangeroepen procedure wordt uitgevoerd, worden de eventuele commando's na DEACTIVATE niet uitgevoerd (het commando RETURN wordt namelijk onmiddellijk daarna uitgevoerd). Als ON PAD is opgegeven, maakt DEACTIVATE MENU ook alle bijbehorende pop-up en andere menu's inactief.

Voorbeeld

In het volgende voorbeeld wordt een balkmenu met twee opties gedefinieerd. In dit menu kunt u met de ene optie een overzicht van een bestandsindex weergegeven en met de andere optie het menu inactief maken.

```
. DEFINE MENU Test MESSAGE "Test"
. DEFINE PAD Dir OF Test PROMPT "Bestandsindex" AT 0,0
. DEFINE PAD Inac OF Test PROMPT "Inactiveren" AT 0,15
. ON SELECTION PAD Dir OF Test DIR
. ON SELECTION PAD Inac OF Test DEACTIVATE MENU
. ACTIVATE MENU Test
```

Zie ook

ACTIVATE MENU, CLEAR MENUS, DEFINE MENU, ON SELECTION, RELEASE MENUS

DEACTIVATE POPUP

Met het commando DEACTIVATE POPUP wordt het actieve pop-up menu van het scherm verwijderd, maar niet uit het geheugen gewist. Eventuele tekst die door het pop-up menu werd bedekt, wordt weer weergegeven.

Syntaxis

DEACTIVATE POPUP

Gebruik

DEACTIVATE POPUP heeft geen effect, waaneer het vaaf de commandostip wordt uitgevoerd, omdat dit commando alleen betrekking op het actieve pop-up menu kan hebben. Als u op de toets **Esc** drukt, slaat u dit commando over, maakt u het pop-up menu inactief en keert u terug naar de commandostip of het programma.

Door DEACTIVATE POPUP wordt de besturing overgedragen aan de programmaregel die onmiddellijk volgt op de regel waarmee de popup werd geactiveerd. Als met het commando ON SELECTION een procedure is aangeroepen die zelf weer het commando DEACTIVATE POPUP bevat, worden de eventuele commando's in de procedure die op het commando DEACTIVATE volgen, niet uitgevoerd en keert u met RETURN terug naar de regel die onmiddellijk volgt op ACTIVATE POPUP.

Voorbeeld

In het volgende voorbeeld wordt het pop-up menu Stop_pop dat twee opties heeft, getoond. Met beide opties wordt een procedure opgeroepen die de keuze van de gebruiker verwerkt. Zoals u ziet is een van de opties van het procedurebestand het inactief maken van het pop-up menu.

```
. DEFINE POPUP Stop_pop FROM 3,38
. DEFINE BAR 1 OF Stop_pop PROMPT "Stop"
. DEFINE BAR 2 OF Stop_pop PROMPT "Terug naar commandostip"
. ON SELECTION POPUP Stop_pop DO Stop
. ACTIVATE POPUP Stop_pop

PROCEDURE Stop
DO  CASE
    CASE BAR() = 1
        QUIT
    CASE BAR() = 2
        DEACTIVATE POPUP
ENDCASE
```

Zie ook

ACTIVATE POPUP, CLEAR POPUPS, DEFINE POPUP, ON SELECTION POPUP, RELEASE POPUPS, RESTORE SCREEN, SAVE SCREEN

DEACTIVATE WINDOW

Met het commando DEACTIVATE WINDOW worden de opgegeven vensters inactief gemaakt en van het scherm verwijderd, maar niet uit het geheugen gewist.

Syntaxis

DEACTIVATE WINDOW <namenlijst vensters>/ALL

Gebruik

Met dit commando worden de vensters uit de namenlijst van de vensters inactief gemaakt door deze van het scherm te verwijderen. De vensters worden niet uit het geheugen gewist zodat u deze weer met het commando ACTIVATE WINDOW op het scherm kunt weergeven.

Wanneer u met DEACTIVATE een venster inactief maakt, wordt het venster dat daarvoor met ACTIVATE was geactiveerd weer actief. Als u met de optie ALL van het commando DEACTIVATE alle vensters inactief maakt, hebt u weer de beschikking over het volledige scherm.

Zie ook

ACTIVATE SCREEN, ACTIVATE WINDOW, DEFINE WINDOW, MOVE WINDOW, RESTORE WINDOW, SAVE WINDOW

DEBUG

Met het commando DEBUG hebt u toegang tot de debug-voorziening van dBASE IV.

Syntaxis

DEBUG <bestandsnaam>/>procedurenaam> [WITH <parameterlijst>]

Gebruik

Net als met DO wordt met dit commando een programma of een procedure gecompileerd en uitgevoerd, maar wordt tevens de schermgrote debug-voorziening van dBASE IV opgeroepen.

Het debug-scherm bestaat uit vier vensters die u informatie bieden over de status van het .prg-bestand en de geopende bestanden. In deze vensters kunt u een programma of een procedure uitvoeren en de commando's tijdens de uitvoering zien. U kunt dan het programma of de procedure bewerken, onderbrekingspunten instellen om de uitvoering van het programma te onderbreken en de resultaten van uitdrukkingen weer te geven terwijl het programma wordt uitgevoerd.

Het scherm is verdeeld in een debug-venster, een bewerkvenster, een onderbrekingspuntvenster en een weergavevenster.

In het debug-venster dat zich onderaan het scherm bevindt, wordt het actieve werkgebied, het database-bestand, het programmabestand, de procedure, het record-nummer, het regelnummer, het hoofdindexbestand en de aanwijzing **OPDRACHT**: weergegeven. Wanneer u op **Esc** of op **Ctrl-End** drukt, keert u vanuit elk venster terug naar de aanwijzing **OPDRACHT**:.

Als u bij de aanwijzing **OPDRACHT**: een E typt, wordt het bewerkvenster het actieve venster. In dit venster, dat zich bovenaan het scherm bevindt, wordt het programma dat wordt uitgevoerd of de procedure die wordt uitgevoerd, weergegeven. Wanneer het bewerkvenster het actieve venster is, hebt u toegang tot de tekstverwerker van dBASE IV en kunt u wijzigingen in het programma aanbrengen. U moet het bestand opslaan zodat de wijzigingen behouden blijven. Nadat de wijzigingen zijn aangebracht, zet de debug-voorziening de uitvoering vanuit het oude bestand voort. De debug-voorziening kan een andere commandoregel uitvoeren dan u had verwacht, afhankelijk van de mate waarin u het bestand hebt gewijzigd.

Als u bij de aanwijzing **OPDRACHT**: een B typt, wordt het onderbrekingspuntvenster aan de rechter zijde van het scherm het actieve venster. In het onderbrekingspuntvenster kunt u één of meer voorwaarden invoeren die worden verwerkt nadat elke regel met code is uitgevoerd. Als het resultaat van een van de voorwaarden waar (.T.) is, wordt het programma onderbroken en wordt het debug-scherm weer weergegeven. Druk op **Esc** of **Ctrl-End** als u vanuit het onderbrekingspuntvenster naar **OPDRACHT**: wilt terugkeren.

Als u bij de aanwijzing **OPDRACHT**: een D typt, wordt het weergavevenster aan de linker zijde van het scherm het actieve venster. Links in dit venster kunt u dBASE-uitdrukkingen invoeren. De resultaten worden rechts in het weergavevenster weergegeven.

Bij de aanwijzing **OPDRACHT**: kunt u ook de volgende letters typen:

- L Line (regel) - hiermee geeft u de regel op die daarna moet worden uitgevoerd.
- N Next (volgende) - hiermee geeft u opdracht het volgende commando in de actieve procedure uit te voeren en daarna naar de aanwijzing **OPDRACHT**: terug te keren. Als het commando DO zich in de actieve procedure bevindt, wordt de opgeroepen procedure buiten de debug-omgeving uitgevoerd maar het onderbrekingspunt blijft echter geldig. Als u N door een getal laat voorafgaan, geeft u de debug-voorziening opdracht dat aantal commando's in de actieve procedure uit te voeren voordat wordt teruggegaan naar de aanwijzing **OPDRACHT**:.
- P Program Trace (programmacontrole) - hiermee geeft u de gegevens van de programmacontrole weer, waaronder het actieve programma, de actieve procedure en het actieve regelnummer.
- Q Quit (stop) - hiermee stopt u de debug-voorziening en annuleert u het programma.
- R Run (uitvoeren) - hiermee geeft u opdracht het programma uit te voeren totdat een onderbrekingspunt of een fout wordt aangetroffen.
- S Step (stap) - hiermee geeft u opdracht het volgende commando uit te voeren en daarna naar de aanwijzing **OPDRACHT**: terug te keren. Als u S door een getal laat voorafgaan, geeft u de debug-voorziening opdracht dat aantal commando's stapsgewijs te doorlopen voordat naar de aanwijzing **OPDRACHT**: wordt teruggekeerd. In tegenstelling tot N worden de procedures die vanuit de actieve procedure worden opgeroepen, binnen de debug-omgeving uitgevoerd.
- U Exit (verlaten) - hiermee verlaat u het programma en keert u terug naar de commandostip. Vanaf de commandostip kan de besturing door RESUME weer aan de debug-voorziening worden overgedragen.

  U kunt slechts één DEBUG-sessie tegelijk uitvoeren. U kunt geen tweede sessie starten wanneer de eerste tijdelijk is onderbroken. Als u een DEBUG-sessie verlaat, kunt u met het commando RESUME de tijdelijk onderbroken sessie voortzetten. Als u een tweede DEBUG-commando invoert terwijl de actieve sessie tijdelijk is onderbroken, wordt RESUME uitgevoerd en keert u terug naar de DEBUG-sessie.
- ↵ - hiermee geeft u opdracht 1S of 1N uit te voeren. Als de laatst uitgevoerde opdracht S (stap) was, wordt met ↵ 1S uitgevoerd. Als de laatst uitgevoerde opdracht N (volgende) was, wordt met ↵ 1N uitgevoerd. De standaardinstelling voor ↵ is het uitvoeren van 1S.

F1 Hulp en **F9 Zoom** zijn aan-/uitschakelaars:

U kunt altijd met **F1 Hulp** het paneel Hulp op het scherm weergeven of daarvan verwijderen. In dit paneel bevindt zich een korte omschrijving van de debug-commando's die u achter de aanwijzing **OPDRACHT**: kunt invoeren.

Met **F9 Zoom** worden de debug-vensters van het scherm verwijderd en worden de schermgegevens die zich daaronder bevinden, weergegeven. Wanneer u daarna nogmaals op F9 Zoom drukt, worden de debug-vensters weer op het scherm weergegeven.

Opties

Met de WITH-parameter kunt u op dezelfde manier als met DO parameters overdragen. In de parameterlijst kunt u alle geldige dBASE IV-uitdrukkingen opnemen. Denk eraan dat in de <parameterlijst> bij DEBUG maximaal acht constanten kunnen worden opgenomen, inclusief een bestandsnaam. Het aantal variabelen is onbeperkt.

Zie ook

COMPILE, DO, LINENO(), MODIFY COMMAND, PROGRAM(), SET DEBUG, SET ECHO, SET PROCEDURE, SET STEP, SET TALK, SET TRAP

DECLARE

Met DECLARE worden eendimensionale of tweedimensionale reeksen met geheugenvariabelen gemaakt.

Syntaxis

DECLARE <reeksnaam 1>[{<aantal rijen>,}
<aantal kolommen>]{,<reeksnaam 2>[{<aantal rijen>,}
<aantal kolommen>] ...}

In dit model bevinden de optionele onderdelen zich tussen accolades. De rechte haken zijn verplichte onderdelen van de syntaxis van het commando DECLARE.

Standaardinstellingen

Als het commando DECLARE vanaf de commandostip wordt ingevoerd, is de reeks globaal en als het commando zich in een programmabestand bevindt, is de reeks lokaal. Met het commando PUBLIC kunt u in een programmabestand een globale reeks maken. Voorbeeld:

```
PUBLIC ARRAY Deel[6,2]
```

Als dit commando in een programmabestand wordt gebruikt, wordt hiermee een globale reeks gemaakt met de naam Deel. Voorbeeld:

```
DECLARE Deel[6,2]
```

Als dit commando in een programmabestand wordt gebruikt, wordt een lokale reeks gemaakt met de naam Deel. Als u het commando vanaf de commandostip invoert, is de reeks globaal.

Gebruik

De lijst met reeksdefinities bestaat uit de reeksnamen en de reeksomvang. Net zoals geheugenvariabelen is de lengte van reeksnamen maximaal tien tekens. De reeksnamen kunnen bestaan uit letters, cijfers en onderstrepingstekens. Het eerste teken moet een letter zijn en er mogen geen spaties in voorkomen. De reeksnaam mag gelijk zijn aan een dBASE IV-commando, maar dit kan echter onvoorspelbare resultaten tot gevolg hebben.

De reeksomvang bestaat uit een of twee getallen tussen rechte haken. Het eerste getal is het aantal rijen in de reeks. Het tweede is het aantal kolommen in de reeks. Een rij in een reeks komt overeen met een record in een database en een kolom in een reeks komt overeen met een veld in een database. Als er één getal wordt gebruikt, is de reeks eendimensionaal. Als er twee worden gebruikt, worden de getallen door komma's gescheiden en is de reeks tweedimensionaal. Het maximum aantal dimensies is twee. De maximum grootte van een reeks (rijen x kolommen) is 1170. Voorbeeld:

```
. DECLARE Kosten[15]
```

Hiermee wordt een eendimensionale reeks (een rij) gemaakt.

Wanneer u alleen het aantal kolommen in een rij opgeeft, is dit gelijk aan het definiëren van een reeks met één rij. Bijvoorbeeld:

```
DECLARE Kosten[1,15]
```

De reeks met de naam Kosten bevat vijftien elementen. De nummering van de elementen begint bij 1.

```
. DECLARE Artikelen[8,3]
```

Hiermee wordt de tweedimensionale reeks Artikelen gemaakt die bestaat uit acht rijen en drie kolommen. De reeks bevat 24 elementen die aan de hand van de rij- en kolompositie zijn genummerd.

Met het commando DECLARE wordt een set geheugenvariabelen gemaakt die aanvankelijk allemaal de waarde logisch onwaar (.F.) bevatten. Het gegevenstype van reekselementen wordt pas bepaald wanneer er gegevens in worden opgeslagen. Bijvoorbeeld:

```
. STORE {15-6-88} TO Mdatum[2,2]
```

Hiermee wordt het element Mdatum[2,2] met een datumwaarde geïnitialiseerd, net zoals met:

```
. mdatum[2,2]={15-6-88}
```

Een reeks kan elementen van verschillende gegevenstypen bevatten.

OPMERKING
Elke reeks wordt, ongeacht de gecompliceerdheid, als een geheugenvariabele verwerkt. Dit betekent dat u een reeks kunt vervangen door een variabele met dezelfde naam zonder dat een waarschuwing wordt weergegeven en zonder dat de vervanging moet worden bevestigd. Wees dus voorzichtig wanneer u geheugenvariabelen met namen die gelijk zijn aan de reeksnamen in het geheugen, maakt of vervangt.

De reeksnaam gebruikt één geheugenlokatie van dezelfde geheugenruimte die ook door de andere geheugenvariabelen wordt gebruikt. De reekselementen gebruiken echter geen geheugenlokaties van deze geheugenruimte maar worden opgeslagen in een afzonderlijk blok dat voor de elementen is gereserveerd. Als een te definiëren reeks groter is dan de beschikbare geheugenruimte, wordt de foutmelding **Onvoldoende geheugen** weergegeven. Nadat een reeks is gedefinieerd, worden de elementen op dezelfde manier behandeld als geheugenvariabelen. Naar de elementen wordt verwezen met de betreffende reeksnaam en positie in de reeks, van links naar rechts en van boven naar beneden. Kosten[4] of Artikelen[5,2] zijn bijvoorbeeld namen van elementen uit het voorbeeld.

Alle commando's en functies die u bij geheugenvariabelen kunt gebruiken, kunt u ook bij reekselementen gebruiken mits de reeks is gedefinieerd en het element zich in het bereik van de reeksdefinitie bevindt. Een aantal commando's zoals COPY, APPEND en REPLACE heeft een speciale vorm (COPY TO ARRAY, APPEND FROM ARRAY en REPLACE FROM ARRAY) voor het verwerken van reeksen. Commando's waarmee geheugenvariabelen worden verwerkt (zoals CLEAR ALL, CLEAR MEMORY, LIST/DISPLAY MEMORY, RELEASE, RESTORE en SAVE), ondersteunen zowel reeksen als geheugenvariabelen.

DECLARE

Als u naar niet-bestaande reekscoördinaten verwijst, wordt de foutmelding **Onjuiste array-afmetingen** weergegeven. Als u naar een reeksnaam verwijst die niet met DECLARE is gedefinieerd, wordt de melding **Geen array** weergegeven.

Voorbeelden

In het volgende voorbeeld wordt van het database-bestand Transact het aantal orders en de som van het veld Totaal_rek in de reeks Specificat opgeslagen:

```
. SET TALK OFF
. USE Transact
. DECLARE Specificat[2]
. CALCULATE CNT(), SUM(Totaal_rek) TO ARRAY Specificat
. ? LTRIM(STR(Specificat[1],8,0)), "orders voor ƒ" + LTRIM(STR(Specificat[2],9,2))
12 orders voor ƒ10080,00
```

In het volgende voorbeeld wordt hetzelfde database-bestand in een programmabestand gebruikt. Voor het uitsplitsen van de orders op Klant_nr worden twee reeksen gebruikt:

```
USE Transact ORDER Klant_nr
DECLARE Orders[12,3]                && Gebruik één rij voor iedere klant.
DECLARE Specificat[2]
Mtel = 1
DO WHILE .NOT. EOF()
    Orders[Mtel,1] = Klant_nr       && Klant_nr opslaan.
    CALCULATE CNT(), SUM(Totaal_rek) TO ARRAY Specificat;
        WHILE Klant_nr = Orders[Mtel,1]
    Orders[Mtel,2] = Specificat[1]  && Aantal orders opslaan.
    Orders[Mtel,3] = Specificat[2]  && Totaal per Klant_nr opslaan.
    Mtel = Mtel + 1
ENDDO
? "Klant NR", "Orders" AT 12, "Totaal" AT 24 && Koptekst voor kolommen.
Mklanten = Mtel - 1
Mtel = 1
DO WHILE Mtel <= Mklanten
    ? Orders[Mtel,1], STR(Orders[Mtel,2],8,0) AT 10,;
        STR(Orders[Mtel,3],9,2) AT 20
    Mtel = Mtel + 1
ENDDO
```

Zie ook

APPEND FROM ARRAY, AVERAGE, CALCULATE, CLEAR ALL, CLEAR MEMORY, COPY TO ARRAY, COUNT, LIST/DISPLAY MEMORY, PUBLIC, RELEASE, REPLACE FROM ARRAY, RESTORE, SAVE, SUM

In *Aan de slag met dBASE IV* treft u informatie aan over het toewijzen van geheugenruimte.

DEFINE BAR

Met het commando DEFINE BAR definieert u een balk in een pop-up menu.

Syntaxis

DEFINE BAR <regelnummer> OF <naam pop-up menu> PROMPT <Tuitdr>
[MESSAGE <Tuitdr>] [SKIP [FOR <voorwaarde>]]

Gebruik

Een balk is een aanwijzing of optie in een pop-up menu. DEFINE BAR is alleen mogelijk als u de opties PROMPT FIELD, PROMPT FILES of PROMPT STRUCTURE van het commando DEFINE POPUP niet opgeeft, omdat de opties van PROMPT de plaats innemen van balken en een gedefinieerd pop-up venster vullen.

Geef de regelnummers alleen in positieve gehele getallen op. Cijfers achter de komma worden weggelaten.

Wanneer u een tweede balkaanwijzing definieert voor een regel die al van een balkaanwijzing is voorzien, komt de nieuwe balkaanwijzing in plaats van de oude.

Wanneer u een balk definieert voor een regelnummer dat hoger is dan het totale aantal regels in het pop-up venster, verschuiven de aanwijzingen verticaal binnen het venster.

Wanneer er een waarde ontbreekt aan een BAR, blijft de desbetreffende rij in het pop-up menu blanco en wordt deze overgeslagen door de selectiebalk.

Wanneer de lengte van de balkaanwijzing groter is dan de lengte van de regel in het pop-up menu, wordt de aanwijzing afgekapt. Horizontaal verschuiven in een pop-up venster is niet mogelijk.

U moet tenminste één balk definiëren voor een pop-up venster. Als u dat niet doet, is het pop-up venster leeg en kan het niet worden geactiveerd.

De uitdrukking MESSAGE wordt gecentreerd weergegeven op de onderste regel van het scherm, buiten het pop-up venster. Deze melding van DEFINE BAR komt in de plaats van eventuele meldingen die zijn gedefinieerd met het commando DEFINE POPUP. De melding mag niet langer zijn dan 79 tekens. Resterende tekens worden weggelaten. De melding wordt gekoppeld aan de balkaanwijzing waarmee de melding is gedefinieerd.

Als SET STATUS op ON is ingesteld, verschijnt de melding onder aan het scherm wanneer de cursor in het pop-up venster zich in de balkaanwijzing bevindt die met hetzelfde commando DEFINE BAR is gedefinieerd. Als SET STATUS op OFF is ingesteld, wordt de lokatie van de melding bepaald door de AT-clausule van het commando SET MESSAGE.

Aan elke balkaanwijzing kan een melding van 79 tekens of minder worden gekoppeld.

Gebruik de optie SKIP wanneer een balkaanwijzing wel moet worden weergegeven, maar door de selectiebalk moet worden overgeslagen. Gebruik de optie SKIP FOR wanneer een balkaanwijzing alleen moet kunnen worden gekozen als de voorwaarde FOR waar (.T.) is.

Voorbeeld

Met de volgende regels worden de keuzemogelijkheden voor het pop-up menu Visie_pop gedefinieerd:

```
. Mwijzig = .F.
. DEFINE POPUP Visie_pop FROM 3,4 TO 8,19
. DEFINE BAR 1 OF Visie_pop PROMPT "Record toevoegen"
. DEFINE BAR 2 OF Visie_pop PROMPT "Bewerken"
. DEFINE BAR 3 OF Visie_pop PROMPT REPLICATE("-",16) SKIP
. DEFINE BAR 4 OF Visie_pop PROMPT "Wissen" SKIP FOR Mwijzig
. ACTIVATE POPUP Visie_pop
```

BAR 3 is een horizontale lijn die dient om de keuzemogelijkheid Wissen te scheiden van de keuzemogelijkheden Toevoegen en Bewerken. De optie SKIP zorgt ervoor dat de scheidslijn niet kan worden gekozen. Met behulp van een logische geheugenvariabele (Mwijzig) kunt u de keuzemogelijkheid Wissen uitsluitend gebruiken als Mwijzig logisch onwaar (.F.) is. De selectiebalk kan niet op balk 4 worden geplaatst zolang Mwijzig waar is, maar wel wanneer Mwijzig logisch onwaar (.F.) is.

Zie ook

ACTIVATE POPUP, BAR(), DEACTIVATE POPUP, DEFINE POPUP, ON SELECTION POPUP, POPUP(), PROMPT(), SET MESSAGE, SET STATUS, SHOW POPUP

DEFINE BOX

Met het commando DEFINE BOX wordt een kader rond tekstregels gedefinieerd.

Syntaxis

DEFINE BOX FROM <kol> TO <kol>
HEIGHT <Nuitdr> [AT LINE <regel>] [SINGLE/DOUBLE / <lijnspecificatie>]

Gebruik

Met dit commando kunt u een kader definiëren rond rapporttekst die u wilt benadrukken.

U definieert de eerste kolom links, de laatste kolom rechts, de eerste regel voor de bovenkant van het kader en de hoogte van het kader. Wanneer AT LINE niet is opgegeven, begint het kader op de actieve regel.

De tekenreeks in de kaderdefinitie volgt dezelfde regels als het commando SET BORDER. U kunt een lijst met tekencodes opgeven voor het kader, zoals wordt beschreven onder SET BORDER. Standaard wordt het kader getekend als een enkele lijn. De optie PANEL van SET BORDER wordt niet ondersteund.

Een kader kan alleen worden afgedrukt als de systeemgeheugenvariabele _box op waar (.T.) is ingesteld. Het kader wordt afgedrukt wanneer _pline gelijk is aan de bovenste rij van het kader. Het kader maakt deel uit van de gegevensstroom van dBASE.

Voorbeeld

In het volgende programma worden van de bovenste en de onderste regel 51 tekens overschreven met de functie SPACE(). Hierdoor worden alleen de hoeken afgedrukt. Met de toewijzingsopdracht onder SCAN wordt ervoor gezorgd dat het kader (_box = .T.) alleen om de eerste en de laatste drie records wordt afgedrukt, en verder niet.

```
*Kader.prg
SET TALK OFF
USE Magazijn ORDER Artikel
DEFINE BOX FROM 10 TO 70 HEIGHT 8 DOUBLE
_box = .T.
?? SPACE(51) AT 15
?
Cnt = 1
SCAN
    _box = (Cnt < 4 .OR. Cnt > (RECCOUNT() - 3))
    ?? Artikel AT 12, Omschrijv
    ?
    Cnt = Cnt + 1
ENDSCAN
?? SPACE(51) AT 15
?
```

Het resultaat van deze code is als volgt:

```
BOEKENKAST                TEAKHOUT,2 PLANKEN
BUREAU,DIRECTIE,5 POTEN   EIKENHOUT,FINEER
DOSSIERKAST,2 LADEN       METAAL,BRUIN
DOSSIERKAST,4 LADEN       METAAL,BRUIN
LADENKAST,2 LADEN         METAAL,ZWART
LAMP,STAAND               MESSING,6 POTEN,ENGELS
LAMP,STAAND               MESSING,6 POTEN,ART DECO
LAMP,STAAND               MESSING,6 POTEN,ENGELS
SOFA,6 POTEN              LEDER,BRUIN,HOGE RUGLEUNING
SOFA,6 POTEN              FLUWEEL,GRIJS,FRANS
SOFA,8 POTEN              FLUWEEL,BLAUW,FRANS
STOEL,BUREAU              LEDER,BRUIN,HOGE RUGLEUNING
STOEL,BUREAU              LEDER,BRUIN,HOGE RUGLEUNING
STOEL,BUREAU              LEDER,BRUIN
STOEL,BUREAU              LEDER,BRUIN
STOEL,KANTINE             PLASTIC,GRIJS
TAFEL,BIJZET              EIKENHOUT,2 POTEN,VIERKANT

Commando  D:\...voorbld\MAGAZIJN  Rec EOB/18  Bestan
```

Afbeelding 2-2 Uitvoer in een kader

Zie ook

@...TO, SET BORDER, _box

In hoofdstuk 1, “Basisbegrippen”, wordt beschreven hoe u de uitvoer naar de printer kunt bepalen met behulp van systeemgeheugenvariabelen.

DEFINE MENU

Gebruik het commando DEFINE MENU in combinatie met het commando DEFINE PAD om een menu te definiëren.

Syntaxis

DEFINE MENU <menunaam> [MESSAGE <Tuitdr>]

Gebruik

Dit commando is de eerste stap bij het maken van een menubalk. Dit commando resulteert op zichzelf niet in een menubalk, maar kent alleen een naam toe aan een menubalk en koppelt eventueel een melding aan de menunaam.

Vermeld deze naam van de menubalk bij het commando DEFINE PAD om de menustrips en de bijbehorende meldingen te definiëren.

Als SET STATUS op ON is ingesteld, wordt het optionele tekstveld MESSAGE onder aan het scherm gecentreerd weergegeven. Als dat niet het geval is, wordt de lokatie bepaald door de AT-clausule van het commando SET MESSAGE. De melding mag niet langer zijn dan 79 tekens. Resterende tekens worden weggelaten.

Elke menustrip kan een eigen melding hebben, maar ook kan voor alle keuzemogelijkheden in de menubalk één melding worden gebruikt. Wanneer een melding wordt toegekend aan een PAD, komt deze in plaats van de melding die is opgegeven met DEFINE MENU.

Voorbeeld

```
. DEFINE MENU Hoofd
```

Zie ook

ACTIVATE MENU, DEACTIVATE MENU, DEFINE PAD, MENU(), ON PAD, ON SELECTION PAD, PAD(), SET MESSAGE, SET STATUS

DEFINE PAD

Het commando DEFINE PAD dient om een afzonderlijke strip in een menubalk te definiëren. Als een menu meerdere strips moet bevatten, herhaalt u het commando met dezelfde menunaam totdat alle strips zijn gedefinieerd.

Syntaxis

DEFINE PAD <stripnaam> OF <menunaam> PROMPT <Tuitdr>
[AT <rgl>,<kol>] [MESSAGE <Tuitdr>]

Gebruik

Met dit commando worden de strips voor een gegeven menubalk gedefinieerd. Voor de naam van een strip gelden dezelfde regels als voor de naam van een veld of een alias. Wanneer u een bestaande stripnaam gebruikt bij het definiëren van een nieuwe strip, vervalt de reeds bestaande eerste strip.

De optie <menunaam> moet eerder zijn gedefinieerd met het commando DEFINE MENU.

De PROMPT-tekst wordt weergegeven binnen het menu. Menu-aanwijzingen kunnen overal op het scherm worden geplaatst. Met de (optionele) beeldschermcoördinaten wordt het beginpunt van de tekst op het scherm gedefinieerd. U kunt een verticale menubalk maken door een paar kolomcoördinaten te gebruiken en de rijcoördinaten voor iedere strip te verhogen, of gebruikmaken van een pop-up.

Als u geen coördinaten opgeeft, verschijnt de eerste aanwijzing in de linker bovenhoek van het scherm. Alle daaropvolgende aanwijzingen worden op dezelfde regel geplaatst, telkens één spatie na het einde van de vorige aanwijzing. Zet SET SCOREBOARD op OFF om te voorkomen dat de informatie van SCOREBOARD over de menustrips op de eerste regel heen wordt geschreven.

Om tussen de aanwijzingen te kiezen drukt u op de toetsen → en ←.

Met de optie MESSAGE definieert u een melding en koppelt u deze aan de strip. De melding is maximaal 79 tekens lang. Resterende tekens worden weggelaten. Als SET STATUS op ON is ingesteld, wordt de melding onder aan het scherm gecentreerd weergegeven wanneer de cursor op de bijbehorende strip staat. Als dat niet het geval is, wordt de lokatie bepaald door de AT-clausule van het commando SET MESSAGE. De melding heeft een hogere prioriteit dan een eventuele andere melding die is gedefinieerd met het commando DEFINE MENU.

Het totale aantal strips dat u kunt definiëren wordt alleen beperkt door de beschikbare hoeveelheid geheugen.

Voorbeeld

Voordat u een strip definieert, moet u eerst het menu definiëren. Nadat u de strips hebt gedefinieerd, definieert u de pop-up menu's voor de verschillende strips en vervolgens de balken die in de menu's moeten komen te staan. Met het volgende programma wordt een menu gemaakt met de naam Hoofd, alsmede de vier bijbehorende strips, met pop-up vensters voor elke strip en een balk voor het eerste pop-up venster.

```
* Een menu met de naam Hoofd maken en de strips definiëren

DEFINE MENU Hoofd
DEFINE PAD Visie OF Hoofd PROMPT "Toevoegen/Bewerken" AT 2,4
DEFINE PAD Ganaar OF Hoofd PROMPT "Ganaar/Zoek" AT 2,25
DEFINE PAD Afdrukken OF Hoofd PROMPT "Afdrukken" AT 2,38
DEFINE PAD Stop OF Hoofd PROMPT "Stop" AT 2,50

* Pop-up vensters aan strips toekennen

ON PAD Visie OF Hoofd ACTIVATE POPUP Visie_pop
ON PAD Ganaar OF Hoofd ACTIVATE POPUP Ganaar_pop
ON PAD Afdrukken OF Hoofd ACTIVATE POPUP Afdruk_pop
ON PAD Stop OF Hoofd ACTIVATE POPUP Stop_pop

* Eerste pop-up venster en eerste balk van de strip Visie definiëren

DEFINE POPUP Visie_pop FROM 3,4 MESSAGE "Visie selecteren"
DEFINE BAR 1 OF Visie_pop PROMPT "Kies visie"

* Bekijk het resultaat

ACTIVATE MENU Hoofd
```

Het menu wordt opgeroepen met het commando ACTIVATE MENU.

Zie ook

ACTIVATE MENU, DEFINE MENU, ON PAD, ON SELECTION PAD, PAD (), SET MESSAGE, SET SCOREBOARD, SET STATUS

DEFINE POPUP

Een pop-up menu is een schermvenster met speciale velden, meldingen en een kader. Met het commando DEFINE POPUP definieert u naam, plaats, kader, aanwijzingen en informatieregel van een pop-up venster.

Syntaxis

DEFINE POPUP <naam pop-up menu> FROM <rgl1>,<kol1>
[TO <rgl2>,<kol2>] [PROMPT FIELD <veldnaam>
/PROMPT FILES [LIKE <structuur>]/PROMPT STRUCTURE]
[MESSAGE <Tuitdr>]

Gebruik

De argumenten van het commando DEFINE POPUP werken als volgt:

Voor de naam van een pop-up menu gelden dezelfde regels als voor de naam van een veld of alias. Aan een pop-up menu moet een naam worden toegekend, zodat u het menu kunt oproepen nadat u het hebt gedefinieerd.

Met de coördinaten FROM en TO worden de linker bovenhoek en rechter benedenhoek van het pop-up venster gedefinieerd. Het venster bedekt de eventuele tekst die op die plaats op het scherm wordt weergegeven. Omdat slechts één pop-up actief kan zijn, kunnen meerdere pop-ups worden gedefinieerd met dezelfde beeldschermcoördinaten. Inactief gemaakte pop-ups verdwijnen van het scherm.

De TO-coördinaten hoeven niet te worden opgegeven. Als u deze weglaat, maakt dBASE IV het venster breed genoeg voor het langste veld en hoog genoeg voor het maximale aantal regels. De ruimte voor een pop-up menu op het beeldscherm is beperkt tot de laatste kolom (79) en de regel boven de statusbalk (22). Wanneer de statusbalk niet wordt weergegeven, is de laatste regel de grootte van het beeldscherm min 1 (24 regels op een beeldscherm van 25 regels). De minimale afmetingen van een pop-up moeten zo zijn dat tenminste één kolom en één rij kunnen worden weergegeven.

Wanneer u de TO-coördinaten opgeeft, worden aanwijzingen die te lang zijn, afgekapt. Wanneer niet alle aanwijzingen in het pop-up venster passen, verschuiven ze verticaal binnen het menu als u de cursor verplaatst.

Er zijn drie variaties van de optie PROMPT: PROMPT FIELD, PROMPT FILES en PROMPT STRUCTURE. Als u één van deze drie opties gebruikt bij het definiëren van een pop-up menu, kunt u later het commando DEFINE BAR niet meer gebruiken voor het desbetreffende pop-up menu.

Met de optie PROMPT FIELD wordt de inhoud van het opgegeven veld van alle records in het database-bestand weergegeven in het pop-up venster. U kunt geen memoveld opgeven bij de optie PROMPT FIELD. U kunt echter wel een alias aan een veldnaam laten voorafgaan.

Met ↑ en ↓ kunt u naar de verschillende aanwijzingen in een pop-up gaan. U kunt hiertoe ook de eerste letter van de aanwijzing typen. Als u een bestand wilt selecteren of een aanwijzing wilt uitvoeren, drukt u op ↵.

Met de optie FILES worden de namen van de bestanden die in de actieve catalogus staan, in het pop-up venster weergegeven. Als u de parameter LIKE <structuur> opgeeft, worden alleen de bestanden weergegeven die overeenkomen met de opgegeven structuur. Zonder het filter LIKE <structuur> worden alle bestanden in de actieve catalogus weergegeven.

Met de optie STRUCTURE worden de gedefinieerde velden weergegeven in het pop-up venster. De optie heeft betrekking op alle velden in het actieve database-bestand of, wanneer het commando SET FIELDS op ON is ingesteld, de velden in de SET FIELDS-lijst.

De MESSAGE-uitdrukking wordt gecentreerd weergegeven op de onderste regel van het beeldscherm, buiten het pop-up venster, tenzij u de MESSAGE-uitdrukking een nieuwe positie toewijst met behulp van de AT-clausule van het commando SET MESSAGE. Als SET STATUS op ON is ingesteld, kunt u de MESSAGE-uitdrukking geen nieuwe positie toewijzen, want deze wordt dan altijd gecentreerd weergegeven op de onderste regel van het scherm. De melding mag niet langer zijn dan 79 tekens. Resterende tekens worden weggelaten. Alleen op de informatieregel kan een melding worden gegeven voor de opties FIELD, FILES en STRUCTURE. Deze melding heeft een hogere prioriteit dan meldingen die eventueel zijn opgegeven met het commando SET MESSAGE TO.

Voorbeeld

Hier zijn vier pop-up menu's:

```
. DEFINE POPUP Visie_pop FROM 3,4 TO 8,19
. DEFINE POPUP Ganaar_pop FROM 3,16 TO 6,28
. DEFINE POPUP Afdruk_pop FROM 3,30 TO 7,42
. DEFINE POPUP Stop_pop FROM 3,38 TO 6,57
```

De coördinaten van deze pop-up menu's zijn zodanig, dat de menu's direct onder de bijbehorende strippen op de menubalk van het menu Hoofd in het voorbeeld van het commando DEFINE PAD worden weergegeven.

Zie ook

ACTIVATE POPUP, BAR(), DEFINE BAR, ON SELECTION POPUP, POPUP(), PROMPT(), SET MESSAGE, SET STATUS

DEFINE WINDOW

Met het commando DEFINE WINDOW worden vensters, kaders en beeldschermkleuren voor vensters gedefinieerd.

Syntaxis

DEFINE WINDOW <vensternaam> FROM <rgl1>,<kol1>
TO <rgl2>,<kol2> [DOUBLE/PANEL/NONE/
<lijnspecificatie>] [COLOR [<normaal>] [,<speciaal>] [,<frame>]]

Gebruik

Dit commando dient om de beeldschermcoördinaten en de attributen voor de weergave van het venster en het kader te definiëren. De FROM-coördinaten bepalen de linker bovenhoek van het venster en de TO-coördinaten bepalen de rechter benedenhoek. Telkens als u een venster activeert, wordt gecontroleerd of de coördinaten van dat venster in overeenstemming zijn met het aantal regels dat op het scherm past en of de statusregel wordt weergegeven.

Standaard wordt het kader getekend met een enkele lijn. U kunt het kader ook met dubbele lijnen of in omgekeerde weergave tekenen, of het kader helemaal weglaten. Zie het commando SET BORDER voor meer informatie over kaderdefinitie met behulp van ASCII-codes.

Als u de parameters van SET BORDER verandert, blijft de kaderdefinitie van toepassing die van kracht was op het moment dat het venster werd gedefinieerd. Als geen kleuren worden opgegeven, worden de kleuren gekozen die van kracht zijn op het moment dat het venster wordt gedefinieerd.

Gebruik zo min mogelijk de ASCII-codes 7, 8, 10, 12, 13, 27 en 127 in kaderdefinities. Deze codes worden wel weergegeven op het scherm, maar veroorzaken problemen met de printerstuurprogramma's, of wanneer u op **Shift-PrtSc** drukt om de inhoud van het scherm af te drukken.

Met de optie COLOR kunt u de voorgrond- en achtergrondkleur instellen voor normale en speciale tekens. Met de parameter <frame> kunt u normale en speciale attributen instellen, net als de andere kleurinstellingen, maar dan alleen voor het kader van het venster.

Er kunnen maximaal 20 vensterdefinities tegelijkertijd in het geheugen aanwezig zijn.

Voorbeeld

Dit voorbeeld heeft betrekking op een klein venster in de rechter bovenhoek van het scherm. De kaderdefinitie bevat een asterisk (ASCII 42) voor de randen en het cijfer 1 (ASCII 49) voor de linker bovenhoek als optioneel vensternummer. Voor de andere drie hoeken wordt een plusteken (ASCII 43) gebruikt.

```
. DEFINE WINDOW W1 FROM 1,50 TO 10,79;
CHR(42),CHR(42),CHR(42),CHR(42 ),CHR(49),CHR(43),CHR(43),CHR(43)
. ACTIVATE WINDOW W1
```

Zie ook

ACTIVATE WINDOW, DEACTIVATE WINDOW, RESTORE WINDOW, SAVE WINDOW, SET BORDER TO, SET COLOR, SET STATUS

DELETE

Met het commando DELETE markeert u in het actieve database-bestand records die moeten worden gewist.

Syntaxis

DELETE [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstelling

Tenzij anders wordt bepaald door het bereik of door een FOR- of WHILE-clausule, krijgt alleen het actieve record een wismarkering.

Gebruik

Met dit commando worden records niet uit het database-bestand verwijderd. De records worden alleen voorzien van een wismarkering, en kunnen daarna eventueel met het commando PACK feitelijk uit het bestand worden verwijderd. Wanneer records van een wismarkering zijn voorzien, kunt u de wismarkering ongedaan maken met het commando RECALL.

Wanneer u met de commando's DISPLAY en LIST records oproept, worden de records met een wismarkering aangegeven met een asterisk (*) op de eerste positie van het record.

Bij schermgrote commando's, zoals BROWSE en EDIT, worden de records met een wismarkering aangegeven met **Del** op de statusbalk. In deze modus kunt u met **Ctrl-U** zowel records verwijderen als herstellen. Als SET STATUS op OFF en SET SCOREBOARD op ON is ingesteld, wordt de aanwijzing Del op het scoreboard weergegeven.

Recordwijzer

Wanneer u het commando DELETE geeft, wordt de recordwijzer niet verplaatst, tenzij u een bereik of de clausule FOR of WHILE opgeeft. Daardoor heeft het commando DELETE aan het einde van een bestand, bijvoorbeeld na LIST of DISPLAY ALL, geen effect.

Voorbeeld

Op de volgende wijze wordt record 6 van een wismarkering voorzien:

```
. DELETE RECORD 6
  1 record gewist
. RECALL ALL
  1 record hersteld
```

Zie ook

DELETED(), PACK, RECALL, SET DELETED, SET SCOREBOARD, ZAP

DELETE FILE

DELETE FILE is een andere vorm van het commando ERASE. Zie het commando ERASE in dit hoofdstuk.

DELETE TAG

Met het commando DELETE TAG wist u de labels met de opgegeven namen uit een .mdx-bestand.

Syntaxis

DELETE TAG <labelnaam 1> [OF <.mdx-bestand>]
[,<labelnaam 2> [OF <.mdx-bestand>] ...]

Gebruik

.mdx-bestanden bevatten maximaal 47 labels, die elk een specifieke indexvolgorde kunnen aangeven voor het database-bestand.

Een bijbehorend .mdx-bestand is een .mdx-bestand dat wordt geopend wanneer een database-bestand de actieve database wordt. Deze .mdx-bestanden hebben dezelfde naam als het database-bestand waarbij zij behoren, maar hebben de extensie .mdx. De aanhef van het database-bestand geeft aan dat er een bijbehorend .mdx-bestand is.

Wanneer u in een actief .mdx-bestand één of meer van de indexlabels niet meer nodig hebt, kunt u deze wissen met het commando DELETE TAG. Wanneer u een indexlabel wist, wordt deze volledig uit een .mdx-bestand verwijderd en ontstaat er weer ruimte in het bestand. Met DELETE TAG maakt u dus ruimte voor een nieuwe label in het .mdx-bestand.

Het commando DELETE TAG heeft alleen effect als het .mdx-bestand dat de label bevat, geopend is. De label die wordt verwijderd, hoeft echter niet de actieve index te zijn.

Wanneer u alle labels uit een .mdx-bestand wist, wordt ook het .mdx-bestand zelf verwijderd. Wanneer u dit doet, wordt de aanhef van het database-bestand bijgewerkt om aan te geven dat er niet langer een .mdx-bestand voor het database-bestand aanwezig is. Wanneer er een catalogus is geopend, worden de wijzigingen ook daarin doorgevoerd.

Met de clausule OF kunt u het .mdx-bestand opgeven dat de label bevat. Wanneer de label in een ander .mdx-bestand staat dan het bijbehorende .mdx-bestand, of wanneer twee geopende .mdx-bestanden dezelfde label bevatten, neemt u de clausule OF op in het commando om aan te geven welke label moet worden verwijderd. Wordt een label niet gevonden, dan verschijnt de foutmelding **TAG niet gevonden**.

Speciale gevallen

In een netwerkomgeving kunt u het commando DELETE TAG alleen geven als het database-bestand in de modus EXCLUSIVE is geopend. Is dat niet het geval, dan verschijnt de foutmelding Bestand moet in modus **EXCLUSIVE worden geopend**.

Voorbeeld

Op de volgende wijze verwijdert u de label Klantnaam uit het .mdx-bestand Afnemers:

```
. DELETE TAG Klantnaam OF Afnemers
```

Zie ook

COPY INDEX, COPY TAG, INDEX, MDX(), NDX(), ORDER(), SET INDEX, SET ORDER, TAG(), USE, SET EXCLUSIVE

DIR

Het commando DIR beeldt een lijst af met de database-bestanden in de huidige directory.

Syntaxis

DIRECTORY/DIR [[ON] <station>:] [[LIKE] [<pad>] <structuur>]

Standaardinstelling

Wanneer u geen bestandsnaam of structuur opgeeft, geeft het commando DIR alleen informatie over de database-bestanden die in de bestandsindex staan. Wanneer u geen pad of station opgeeft, geeft het commando DIR informatie over het actieve station en de actieve bestandsindex.

Gebruik

Bij database-bestanden geeft DIR de bestanden weer, het aantal records, de datum van de laatste wijziging, de grootte van het bestand (in bytes), het totale aantal weergegeven bestanden, het totale aantal bytes van de weergegeven bestanden en het resterende aantal bytes in het station. Wanneer u een ander type bestand dan .dbf opgeeft, worden alleen de bestandsnamen weergegeven.

Aan het einde van de padspecificatie van het commando DIR van dBASE IV moet een backslash (\) staan. Als u in dBASE IV de volgende opdracht geeft, verschijnt de melding **Geen**:

```
. DIR \DBASE\VOORBLD
```

Als u dezelfde opdracht vanaf de aanwijzing van DOS geeft, verschijnt een lijst van de bestanden in de bestandsindex Voorbld. Als u deze lijst vanaf de commandostip op het scherm wilt weergeven, moet u de opdracht als volgt typen:

```
. DIR \DBASE\VOORBLD\
```

Voorbeelden

Op de volgende wijze geeft u de database-bestanden in de actieve bestandsindex weer:

```
. DIR
```

Op de volgende wijze geeft u alle bestandsnamen in de actieve bestandsindex weer:

```
. DIR *.*
```

Op de volgende wijze geeft u alle gecompileerde programmabestanden in de subbestandsindex \VERKOOP weer:

```
. DIR \VERKOOP\*.dbo
```

Op de volgende wijze geeft u bestandsnamen weer die drie tot vijf tekens lang zijn en waarvan het derde teken een D is:

```
. DIR ??D??.*
```

Op de volgende wijze geeft u alle .dbf-bestanden in de bestandsindex Verkoop weer:

```
. DIR \VERKOOP\
```

Zie ook

LIST/DISPLAY FILES, SET DEFAULT, SET DIRECTORY

DISPLAY-commando's

De volgende DISPLAY-commando's lijken op hun LIST-tegenhangers. Wanneer u de DISPLAY-versie gebruikt, wordt er slechts zoveel informatie weergegeven als er op het beeldscherm past. Als er dan nog meer informatie is, wordt u gevraagd op een toets te drukken om het volgende scherm zichtbaar te maken.

Wanneer u de optie TO FILE <bestandsnaam> opgeeft, wordt de uitvoer van zowel DISPLAY als LIST in een bestand opgeslagen.

Zie de volgende LIST-commando's voor informatie over zowel de LIST- als de DISPLAY-versie.:

LIST/DISPLAY
LIST/DISPLAY FILES
LIST/DISPLAY HISTORY
LIST/DISPLAY MEMO
LIST/DISPLAY STATUS
LIST/DISPLAY STRUCTURE
LIST/DISPLAY USERS

Met het commando DO voert u een commandobestand of procedure van dBASE IV uit. Als het commandobestand of de procedure nog niet is gecompileerd, volgt er een syntaxiscontrole, waarna het bestand of de procedure wordt gecompileerd en opgeslagen als een objectbestand met de extensie .dbo. Vervolgens wordt het .dbo-bestand uitgevoerd.

Met DO kunnen ook parameters naar het opgegeven programma worden overgebracht.

Syntaxis

DO <naam programmabestand>/<naam procedurebestand>
[WITH <parameterlijst>]

Standaardinstelling

OPMERKING
Alle objectcodebestanden van dBASE IV 1.0 moeten in dBASE IV 1.1 opnieuw worden gecompileerd als u ze in versie 1.1 wilt draaien. In de meeste gevallen vindt het opnieuw compileren automatisch plaats. Door de commando's waarmee objectbestanden van dBASE worden gedraaid, wordt de versie van het codebestand gecontroleerd en vervolgens opnieuw gecompileerd wanneer blijkt dat het codebestand met versie 1.0 is gemaakt en het bijbehorende bronbestand wordt aangetroffen. Door dBASE IV wordt in de actieve bestandsindex en alle bestandsindexen in het dBASE-pad gezocht naar de bronbestanden. Een bestand wordt niet opnieuw gecompileerd als het bronbestand een andere naam heeft gekregen of niet wordt aangetroffen, of als het codebestand met het hulpprogramma DBLINK is gemaakt.

Als het bestand niet op het standaardstation staat of als er geen pad is opgegeven, moet de stationsaanduiding aan de bestandsnaam voorafgaan. Wanneer het bestand niet in de standaardbestandsindex staat en er geen pad is ingesteld met SET PATH, moet ook het pad aan de bestandsnaam voorafgaan.

Voor de namen van procedures moet u echter geen stationsaanduiding of pad opgeven. De stationsaanduiding en het pad voor de procedurebestanden geeft u op met het commando SET PROCEDURE.

De volgende zoekvolgorde is van toepassing als u een DO-commando geeft.

1. DO zoekt naar een procedure in het actieve .dbo-bestand, als er een dergelijk bestand wordt uitgevoerd.
2. DO zoekt naar een procedure in een procedurebestand, als er een dergelijk bestand is geactiveerd met SET PROCEDURE. Nadat een dergelijke procedure is aangetroffen, wordt niet verder gezocht naar andere procedures met dezelfde naam.
3. DO zoekt naar een procedure in andere geopende .dbo-bestanden.
4. DO zoekt naar een .dbo-bestand met de opgegeven naam.
5. DO zoekt naar een .prg-bestand met de opgegeven naam, compileert het en voert het daaruit resulterende .dbo-bestand uit.
6. DO zoekt naar een .prs-bestand met de opgegeven naam, compileert het en voert het daaruit resulterende .dbo-bestand uit.

DO controleert of het bestand een object- of een bronbestand is. Het bestand wordt uitgevoerd als het een objectbestand (.dbo) is, of gecompileerd en uitgevoerd als het een bronbestand (.prg of .prs) is.

Gebruik

dBASE IV ondersteunt het concept van procedures binnen de programma's door een procedurelijst bij te houden in de objectbestanden (.dbo). Wanneer een bronbestand niet begint met het commando PROCEDURE of FUNCTION, wordt de code gecompileerd als procedure en toegevoegd aan de procedurelijst in het objectbestand met dezelfde naam als het bronbestand. Een typisch .prg-bestand zoals:

```
* Centraal.prg
? "CENTRAAL"
RETURN
```

wordt gecompileerd en weggeschreven naar een .dbo-bestand met één procedure, Centraal. Wanneer u het commando DO CENTRAAL geeft, wordt eerst het .dbo-bestand met deze naam gezocht en wordt vervolgens de desbetreffende procedure in het .dbo-bestand gezocht.

Een bronbestand kan meerdere procedures bevatten, bijvoorbeeld:

```
* Centraal.prg
? "CENTRAAL"
DO Perifeer
RETURN

PROCEDURE Perifeer
? "PERIFEER"
RETURN
```

Let erop dat alleen commando's aan het begin van een bestand de standaardprocedurenaam krijgen. Commando's tussen RETURN en PROCEDURE veroorzaken een foutmelding tijdens het compileren. Dergelijke commando's mogen wel in het programmabestand blijven staan, maar worden dan niet uitgevoerd met DO.

Iedere procedure in een actief .dbo-bestand is beschikbaar voor het commando DO. Wanneer A.dbo B.dbo aanroept en B.dbo C.dbo aanroept, zijn alle procedures die zijn gedefinieerd in A, B en C, beschikbaar voor elke procedure in C. dBASE IV ondersteunt het commando SET PROCEDURE TO van dBASE III en dBASE III PLUS, maar het commando is alleen nodig om toegang te verkrijgen tot procedures in een bestand dat niet door DO <bestandsnaam> is geactiveerd.

Er kunnen maximaal 32 .dbo-bestanden actief zijn. Een dbo-bestand is actief als u het hebt geopend met SET PROCEDURE TO, of als de programmabesturing met het commando RETURN aan dat bestand kan worden teruggegeven.

Het totale aantal geopende bestanden, inclusief database-bestanden, indexbestanden, indelingsbestanden en commandobestanden, wordt bepaald door de instelling van FILES in het Config.sys-bestand. Door DOS worden vijf bestanden gereserveerd. U kunt dus maximaal 94 geopende bestanden hebben als u FILES = 99 in het Config.sys-bestand opgeeft. De maximale waarde van FILES is 99.

Wanneer het met DO aangeroepen programma is uitgevoerd, wordt de besturing teruggegeven aan het programma waarin het commando werd gegeven (of aan de commandostip of het Control Center als het commando daar werd gegeven).

Geheugenvariabelen en reeksen die worden gemaakt in het opgeroepen programma, moeten PUBLIC zijn als ze van kracht moeten blijven nadat het opgeroepen programma is beëindigd. Alle PRIVATE geheugenvariabelen en reeksen die in het programma worden gemaakt, gaan verloren wanneer het programma wordt beëindigd.

Omdat DO eerst naar .dbo-bestanden zoekt, mag de extensie van het bestand niet worden gewijzigd nadat het bestand is gecompileerd. Een .dbo-bestand mag ook geen nieuwe naam worden gegeven. Als u namelijk met DO een .dbo-bestand met een nieuwe naam wilt uitvoeren, wordt de foutmelding **Procedure niet gevonden** weergegeven wanneer de hoofdprocedure door dBASE IV niet met de nieuwe naam van het .dbo-bestand kan worden gevonden.

De tekstverwerker van dBASE IV, die kan worden opgeroepen met het commando MODIFY COMMAND of vanuit het Control Center, verwijdert het verouderde .dbo-bestand wanneer een .prg-bestand wordt gewijzigd, en compileert het gewijzigde .prg-bestand opnieuw wanneer dit wordt opgeslagen. Het .dbo-bestand wordt dus onmiddellijk bijgewerkt. Omdat andere tekstverwerkers dit niet doen, kunt u er met SET DEVELOPMENT voor zorgen dat u nooit een verouderd .dbo-bestand gebruikt.

Als SET DEVELOPMENT op ON is ingesteld en u gebruikt een andere tekstverwerker voor het bewerken van bestaande programmabestanden (.prg), vergelijkt DO de tijd en de datum van een bronbestand met de tijd en datum van het bijbehorende .dbo-bestand. Als het .dbo-bestand ouder is dan het bronbestand, compileert DO de broncode opnieuw voordat het commando wordt uitgevoerd.

Opties

Met de optie WITH kunt u parameters aan een procedure toekennen. De parameterlijst mag elke geldige uitdrukking van dBASE bevatten. U kunt maximaal 50 parameters toekennen. Maximaal tien parameters kunnen letterlijke waarden bevatten en maximaal zes kunnen onderdeel uitmaken van een reeks. De overige parameters moeten variabel zijn. Veldnamen hebben een hogere prioriteit dan geheugenvariabelen. Wilt u een geheugenvariabele als parameter toekennen in plaats van een veld met dezelfde naam, plaats dan *M*-> voor de naam van de variabele.

Tips

Vermijd het gebruik van het commando Bestanden (.prg). Een commandobestand kan een DO-commando bevatten dat zichzelf steeds opnieuw uitvoert. Het kan ook voorkomen dat een commandobestand een subroutine oproept, waarna deze subroutine het originele commandobestand weer oproept. In beide gevallen is er sprake van recursie. Dit kan uiteindelijk de foutmelding **Procedure/functie te diep genest** tot gevolg hebben. Standaard kan DO 16 keer worden gebruikt, maar het aantal geneste DO-commando's wordt beperkt door de hoeveelheid geheugenruimte die beschikbaar is.

De besturing moet aan het aanroepende programma worden teruggegeven met behulp van het commando RETURN en niet met het commando DO. Als een commandobestand eigen opdrachten opnieuw moet uitvoeren, moeten die commando's worden opgenomen in een DO WHILE-lus. Het commandobestand mag zichzelf niet opnieuw aanroepen met DO.

Als u het commando DO geeft voor een procedurebestand met een willekeurige extensie, wordt er een .dbo-bestand gemaakt. U moet echter het commando geven dat hoort bij het betreffende bestandstype, bijvoorbeeld REPORT FORM, om het juiste, gecompileerde .dbo-bestand te maken (in dit geval .fro).

Speciale gevallen

Als het invoerbestand de extensie .upd heeft, resulteert DO in een .dbo-bestand en wordt vervolgens de mutatie-query uitgevoerd. U kunt de extensie .dbo wijzigen in .upo als u de mutatie-queries gescheiden wilt houden van de programmabestanden.

Voorbeelden

Het volgende programma, Oppberek.prg, berekent de oppervlakte van een rechthoek aan de hand van de formule oppervlakte = lengte * breedte:

```
* Naam programma: Oppberek.prg
PARAMETERS M_lengte, M_breedte, M_oppervl
M_oppervl = M_lengte * M_breedte
RETURN
* EOP: Oppberek.prg
```

U voert het programma Oppberek uit door bijvoorbeeld de waarden 4 en 6 toe te kennen aan de geheugenvariabelen M_lengte en M_breedte en de uitkomst op te slaan als de geheugenvariabele M_uitkomst:

```
. M_uitkomst = 0
0
. DO Oppberek WITH 6, 4, M_uitkomst
Gecompileerd wordt regel        5
24
. ? M_uitkomst
24
```

Zie ook

CANCEL, COMPILE, CREATE/MODIFY QUERY/VIEW, DEBUG, FUNCTION, MODIFY COMMAND, PARAMETERS, PRIVATE, PROCEDURE, PUBLIC, RESUME, RETURN, SET DEBUG, SET DEVELOPMENT, SET ECHO, SET PROCEDURE, SET TRAP, SUSPEND

DO CASE

DO CASE is een commando voor gestructureerd programmeren, waarmee één mogelijkheid wordt gekozen uit een reeks alternatieven.

Syntaxis

```
DO CASE
    CASE <voorwaarde>
    <commando's>
    [CASE <voorwaarde>
    <commando's>]
    .
    .
    .
    [OTHERWISE
    <commando's>]
ENDCASE
```

Gebruik

ENDCASE vormt het einde van een DO CASE-structuur. Commandoparen zoals DO CASE...ENDCASE, IF...ENDIF en DO WHILE...ENDDO moeten op de juiste manier worden genest binnen een DO CASE-structuur. DO CASE-commando's mogen worden genest.

CASE <voorwaarde> stelt een voorwaarde, of een logische uitdrukking als A = B of Numvar < 11, die moet worden getoetst. Wanneer de getoetste uitdrukking waar is (.T.), worden alle daaropvolgende commando's uitgevoerd tot de eerstvolgende CASE, OTHERWISE of ENDCASE wordt bereikt.

Nadat de bij een CASE behorende commando's zijn uitgevoerd, worden er verder geen CASE-opdrachten meer uitgevoerd. Het programma wordt voortgezet met het eerste commando na ENDCASE. Ook wanneer niet één CASE-voorwaarde waar blijkt te zijn en er geen OTHERWISE-opdracht is opgegeven, wordt het eerste commando na ENDCASE uitgevoerd. OTHERWISE en de bijbehorende commando's worden alleen uitgevoerd als *alle* met CASE opgegeven voorwaarden onwaar (.F.) zijn.

Tips

Slechts één van de mogelijkheden wordt uitgevoerd, ook wanneer meerdere van toepassing zijn. In situaties waarin alleen rekening moet worden gehouden met de eerste voorwaarde die waar (.T.) is, verdient het commando DO CASE de voorkeur boven het commando IF.

De CASE-constructie wordt vaak toegepast wanneer er een paar uitzonderingen zijn op een voorwaarde. De opdrachten CASE <voorwaarde> geven dan de uitzonderingen weer en de opdracht OTHERWISE weerspiegelt de gebruikelijke situatie.

DO CASE

Voorbeeld

Vergelijk dit voorbeeld met het voorbeeld dat is gegeven bij het commando IF. Met de volgende CASE-constructie wordt de grootte van een variabele bepaald en wordt de bijbehorende melding weergegeven:

```
DO CASE
    CASE M_waarde >100
        ? "Waarde is groter dan 100."
    CASE M_waarde >10
        ? "Waarde is groter dan 10."
    CASE M_waarde >1
        ? "Waarde is groter dan 1."
    OTHERWISE
        ? "De waarde is 1 of minder."
ENDCASE
```

Zie ook

DO, DO WHILE, IF, IIF()

DO WHILE

DO WHILE is een commando voor gestructureerd programmeren, waarmee de opdrachten tussen DO WHILE en de bijbehorende ENDDO worden herhaald zolang de opgegeven voorwaarde waar is (.T.).

Syntaxis

```
DO WHILE <voorwaarde>
    <commando's>
    [LOOP]
    [EXIT]
ENDDO
```

Gebruik

DO WHILE <voorwaarde> vormt het begin van een gestructureerde procedure, waarin de eropvolgende commando's alleen worden uitgevoerd als de voorwaarde in de logische waarde waar (.T.) resulteert, bijvoorbeeld .NOT. EOF() .AND. Mvar1 = 11.

Wanneer de voorwaarde waar (.T.) is, worden alle daaropvolgende commando's uitgevoerd tot de eerstvolgende ENDDO, LOOP of EXIT. In het geval van ENDDO en LOOP gaat het programma terug naar het commando DO WHILE, waarna de voorwaarde opnieuw wordt getoetst. In het geval van EXIT wordt het programma voortgezet vanaf de opdracht die na ENDDO staat.

Met het commando LOOP gaat u terug naar het begin van de programmastructuur DO WHILE...ENDDO. Hiermee voorkomt u dat de resterende commando's in de DO WHILE-constructie worden uitgevoerd.

Met het commando EXIT verlaat u de DO WHILE...ENDDO-lus, en wordt het eerste commando na ENDDO uitgevoerd.

ENDDO vormt het einde van een DO WHILE-structuur. De ruimte na ENDDO op de commandoregel kan worden gebruikt voor commentaar. De commentaaraanduiding && is niet noodzakelijk, al wordt er wel een waarschuwing gegeven tijdens het compileren.

Alle gestructureerde commando's in een DO WHILE...ENDDO-structuur moeten op de juiste manier zijn genest. DO WHILE's mogen ook worden genest.

Als de voorwaarde onwaar (.F.) is, worden de commando's tussen DO WHILE en ENDDO overgeslagen en wordt het commando uitgevoerd dat na ENDDO volgt.

Per procedure kunt u in dBASE IV maximaal 32 keer het commando DO WHILE gebruiken.

Richtlijnen voor het programmeren

U kunt alleen macro's gebruiken in het voorwaardelijke deel van een DO WHILE-lus als de waarde van de variabele in de macro niet verandert, omdat de syntaxis van de DO WHILE-opdracht alleen wordt gecontroleerd wanneer de lus voor de eerste keer wordt doorlopen. Daarna wordt de DO WHILE-opdracht vanuit het geheugen uitgevoerd.

DO WHILE

Ook macrosubstitutie kan alleen plaatsvinden op het laagst geneste niveau van een programma en binnen de laagst geneste DO WHILE-lus. Wanneer de DO WHILE-lus met de macro een geneste DO WHILE-lus of het commando DO <procedurenaam> bevat, is de voorwaarde van de lus na de eerste toetsing altijd waar (.T.). Het programma raakt dan in een eindeloze lus.

Voorbeelden

Het eerste voorbeeld laat zien hoe u een macro invoegt in het voorwaardelijke deel van een DO WHILE-lus:

```
* Voorbeeld van een correct gebruikte macro in een DO WHILE-opdracht.
*
USE Transact ORDER Klant_nr
Voorwaarde = [UPPER(Klant_nr) = "C00001"]
FIND C00001
DO WHILE &Voorwaarde. .AND. .NOT. EOF()
    * De waarde van Voorwaarde verandert niet binnen de lus.
    ? Bestel_nr, Dat_trans, Totaal_rek
    SKIP
ENDDO
CLOSE DATABASE
```

Het volgende voorbeeldprogramma laat zien hoe u de optie EXIT gebruikt. Met het programma kunt u vijf records uit het database-bestand Magazijn bekijken en daarna besluiten of u de volgende vijf of de vorige vijf records wilt zien.

```
* Naam programma: Deel.prg
USE Magazijn
DO WHILE .NOT. EOF()
CLEAR
LIST NEXT 5 Bestel_nr, Artikel, Stukprijs
?
WAIT "X = stoppen, V = vorige records, spatiebalk = verder." TO Mstop
DO CASE
    CASE Mstop $ "Xx" .OR. EOF()
        EXIT
    CASE Mstop $ "Vv"
        SKIP -9
    OTHERWISE
        SKIP
ENDCASE
ENDDO
* EOP: Deel.prg
```

Zie ook

&(macro), DO, DO CASE, IF, RETURN, SCAN

EDIT

EDIT is een schermgroot commando voor het weergeven of bewerken van de inhoud van een record in het actieve database-bestand of de actieve visie.

Syntaxis

EDIT [NOINIT] [NOFOLLOW] [NOAPPEND] [NOMENU] [NOEDIT] [NODELETE] [NOCLEAR] [<recordnummer>] [FIELDS <veldenlijst>] [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstelling

Als u het commando EDIT geeft zonder een bereik of een FOR- of WHILE-clausule op te geven, kunt u alle records in het database-bestand bewerken.

Geeft u wel een bereik of een FOR- of WHILE-clausule op, dan worden het menu **Ga naar** en de **F2**-toets buiten werking gesteld, zodat u niet kunt overschakelen op BROWSE.

Gebruik

EDIT en CHANGE zijn identieke commando's.

U kunt van EDIT overschakelen op BROWSE door op **F2 Gegevens** te drukken. Als er een indelingsbestand (.fmt) actief is, worden de gegevens door EDIT weergegeven in de indeling die wordt bepaald door dat indelingsbestand. Is er geen indelingsbestand actief, dan is de standaard verticale veldindeling van toepassing. Bij BROWSE worden meerdere records in kolommen weergegeven.

EDIT maakt gebruik van de normale stuurtoetsen. Met de cursortoetsen verplaatst u de cursor in een record. Met **PgUp** gaat u naar het vorige record en met **PgDn** naar het volgende record of, als de velden niet op één scherm passen, naar het volgende scherm. Druk op **Ctrl-End** om EDIT te verlaten en alle veranderingen op te slaan. Druk op **Esc** om EDIT te verlaten en alle gewijzigde records, behalve het actieve record, op te slaan. Wilt u een memoveld bewerken, dan plaatst u de cursor op de naam van het memoveld en drukt u op **Ctrl-Home** of op **F9 Toevoegen**.

Tenzij u de optie NOAPPEND hebt opgegeven, kunt u met EDIT records toevoegen aan een database-bestand door de cursor voorbij het laatste record in het bestand te verplaatsen. EDIT heeft dan hetzelfde effect als het commando APPEND.

Als EDIT wordt aangeroepen vanuit BROWSE, zijn alle opties van de commandoregel van BROWSE beschikbaar, behalve COMPRESS, FREEZE, LOCK, WIDTH en WINDOW. Omgekeerd gelden voor BROWSE ook de commandoregelopties van EDIT. Voor EDIT is het volledige scherm nodig en kan dus niet in een venster worden gebruikt, ook niet als u dit commando vanuit een BROWSE-venster aanroept door op **F2 Gegevens** te drukken.

Als het commando EDIT is beëindigd, keert u terug naar het uitgangspunt: de commandostip, de volgende regel van een .prg-bestand of het Control Center.

Opties

Met NOINIT kunt u gebruik blijven maken van de opties die u hebt opgegeven bij een eerder EDIT-commando. In dat geval wordt de EDIT-tabel niet geïnitialiseerd, maar wordt de tabel van het laatste EDIT-commando gebruikt.

NOFOLLOW is alleen van toepassing als de records zijn geïndexeerd. Als u NOFOLLOW opgeeft en een sleutelveld in een record wijzigt, krijgt het record een andere plaats in de indexvolgorde en wordt het record dat de plaats inneemt van het oude record, het nieuwe actieve record op het scherm. Als u NOFOLLOW niet opgeeft, wordt het bewerkte record verplaatst nadat de sleutel is gewijzigd, en wordt het record na het verplaatste record het actieve record op het scherm.

Als u NOAPPEND opgeeft, kunnen er geen records aan het database-bestand worden toegevoegd.

Als u NOMENU opgeeft, wordt de menubalk van EDIT uitgeschakeld.

Geeft u NOEDIT op, dan kunnen de gegevens op het scherm niet worden gewijzigd. U kunt echter wel records toevoegen door de cursor voorbij het laatste record te verplaatsen, of records een wismarkering geven.

Als u NODELETE opgeeft, kunnen er geen records worden gewist.

Bij NOCLEAR blijft het record op het scherm zichtbaar nadat u EDIT hebt beëindigd.

Wanneer u <recordnummer> opgeeft, begint u bij het desbetreffende record, maar kunt u nog steeds naar andere records gaan. U kunt ook het sleutelwoord RECORD gebruiken, een van de opties van <bereik>. Als u echter het sleutelwoord RECORD invoegt, kunt u slechts één record bewerken en kunt u niet naar andere records in het bestand gaan. Omdat met EDIT RECORD <recordnummer> slechts het opgegeven record kan worden bewerkt, zijn EDIT RECORD <recordnummer> en EDIT <recordnummer> niet identiek.

Speciale gevallen

Als u in een netwerk vergeten bent een record te vergrendelen met **Ctrl**-O of LOCK() voor u een wijziging aanbrengt, probeert dBASE IV het record en alle gerelateerde records te sluiten zodra u op een toets drukt die geen stuurtoets is.

Stuurtoetsen

In de volgende tabel treft u een overzicht aan van de stuurtoetsen die u bij EDIT kunt gebruiken.

Tabel 2-6 Stuurtoetsen

Toets	Functie
F1	Hulp
F2	BROWSE
F3	Naar einde van record
F4	Naar begin van record
F5-F9	Geen functie
F10	Menu Records
Shift-F2	Naar query-scherm
Shift-F3 - Shift-F7	Geen functie
Shift-F8	Hetzelfde veld uit het vorige record kopiëren
Shift-F9	Menu Afdrukken
Shift-F10	Verwerking van macro's (via toetsenbord)

Zie ook

BROWSE, CHANGE, CREATE/MODIFY QUERY/VIEW, MODIFY COMMAND, SET DESIGN, SET FIELDS, SET FORMAT, SET LOCK, SET REFRESH, SET WINDOW OF MEMO

EJECT

Met het commando EJECT geeft u de printer opdracht het papier door te voeren naar het begin van de volgende pagina. EJECT is alleen van invloed op de printer. Meer mogelijkheden biedt het commando EJECT PAGE.

Syntaxis

EJECT

Gebruik

Tenzij u de systeemvariabele _padvance hebt ingesteld op "LINEFEEDS", zendt u met EJECT een paginadoorvoer (ASCII-code 12) naar de printer. Als _padvance op "LINEFEEDS" is ingesteld, zendt u met EJECT regeldoorvoertekens (ASCII-code 10) naar de printer om naar het begin van de de pagina te gaan.

De printer functioneert alleen naar behoren als u eerst het papier aan het begin van de pagina plaatst. Raadpleeg het boek bij uw printer voor instructies.

Met EJECT worden de waarden PROW() en PCOL() op nul ingesteld.

Tips

Het is raadzaam om in een programmabestand eerst met PRINTSTATUS() te controleren of de printer aangesloten en online is, voordat u het commando EJECT geeft.

Zie ook

???, EJECT PAGE, ON PAGE, PCOL(), PRINT, PRINTSTATUS(), PROW(), SET PRINTER

In hoofdstuk 5, "Systeemgeheugenvariabelen", worden ook de systeemvariabelen _padvance, _pageno, _pcolno en _plineno behandeld.

EJECT PAGE

Met het commando EJECT PAGE verplaatst u de uitvoerstroom naar de gedefinieerde ON PAGE-positie op de actieve pagina, of naar het begin van de volgende pagina.

Syntaxis

EJECT PAGE

Gebruik

Als u het commando ON PAGE hebt gegeven, wordt bij het commando EJECT PAGE gecontroleerd of het actieve regelnummer (_plineno) hoger of lager is dan het regelnummer van ON PAGE.

1. Als _plineno lager is dan het regelnummer van ON PAGE, wordt het aantal regeldoorvoertekens naar de uitvoerapparatuur gezonden dat nodig is om de ON PAGE-positie te bereiken.

2. Als u het commando ON PAGE niet hebt gegeven of als het actieve regelnummer (_plineno) hoger is dan het regelnummer van ON PAGE, heeft EJECT PAGE het volgende resultaat:

   a. Als SET PRINTER op ON is ingesteld en _padvance gelijk is aan "FORMFEED", wordt een paginadoorvoer naar de printer gezonden.

      Als SET PRINTER is ingesteld op ON en _padvance op "LINEFEEDS", wordt het aantal regeldoorvoertekens naar de printer gezonden dat nodig is om de pagina door te voeren. Dit aantal wordt berekend met de formule (_plength -_plineno).

      Als de uitvoer ook een andere bestemming heeft gekregen (bijvoorbeeld met SET PRINTER of SET ALTERNATE), wordt op basis van deze formule (_plength - _plineno) bepaald, hoeveel regeldoorvoertekens moeten worden verzonden.

   b. De waarde van de systeemgeheugenvariabele _pageno wordt verhoogd.

   c. De waarde van de systeemgeheugenvariabelen _plineno en _pcolno wordt op nul ingesteld.

Tips

U kunt het commando EJECT PAGE geven om verder te gaan naar de ON PAGE-positie op de active pagina. ON PAGE zorgt dan verder voor de paginadoorvoer. Eventueel wordt daarbij een voettekst onder aan de actieve pagina en een koptekst boven aan de volgende pagina geplaatst.

Gebruik EJECT in plaats van EJECT PAGE om alleen een paginadoorvoer naar de printer te sturen. Het commando EJECT heeft geen invloed op de systeemvariabelen _pageno en _plineno, maar wordt wel beïnvloed door _padvance en _pwait.

Zie ook

?/??, EJECT, ON PAGE, SET ALTERNATE, SET PRINTER, _padvance, _pageno, _plength, _plineno

In hoofdstuk 1, “Basisbegrippen”, worden de systeemgeheugenvariabelen en de uitvoerstroom behandeld.

ERASE/DELETE FILE

Met ERASE verwijdert u een bestand uit de bestandsindex van een station (DELETE FILE is een andere vorm van ERASE).

Syntaxis

ERASE <bestandsnaam>/?

of

DELETE FILE <bestandsnaam>/?

Standaardinstelling

De bestandsnaam moet van een extensie zijn voorzien. Als het bestand niet op het standaardstation staat, moet ook de stationsaanduiding worden opgegeven.

Gebruik

WAARSCHUWING
Zelfs wanneer SET SAFETY op ON is ingesteld, wordt bij ERASE niet om een bevestiging gevraagd.

Als u het commando ERASE ? geeft, verschijnt er een menu met bestanden.

U kunt een geopend bestand niet verwijderen.

Als u een bestand wilt verwijderen dat in een andere bestandsindex staat, moet u het volledige pad opgeven in de bestandsnaam.

Wanneer u een database-bestand (.dbf) met memovelden verwijdert, moet u het .dbt-bestand waarin de memovelden zijn opgeslagen, apart verwijderen. Denk er ook aan het bijbehorende .mdx-bestand te verwijderen (indien aanwezig).

In tegenstelling tot het commando ERASE van DOS, mogen in dBASE IV geen jokers worden gebruikt.

DELETE FILE is hetzelfde als ERASE.

Zie ook

CLOSE, DELETE, DELETE TAG, FILE(), SET SAFETY, USE

EXPORT

Met het commando EXPORT kopieert u het geopende database-bestand naar een ander bestand met een indeling PFS:FILE, dBASE II, Framework II of RapidFile.

Syntaxis

EXPORT TO <bestandsnaam> [TYPE] PFS/DBASEII/FW2/RPD
[FIELD <veldenlijst>] [<bereik>]
[FOR <voorwaarde>] [WHILE <voorwaarde>]

Gebruik

EXPORT resulteert in bestanden die geschikt zijn voor PFS:FILE, dBASE II, Framework II of RapidFile. Met het commando COPY kunt u bestanden maken die door andere programma's kunnen worden gelezen.

Als er een index actief is, worden de records in de indexvolgorde geëxporteerd. Voor PFS:FILE-bestanden kunt u met behulp van een indelingsbestand (.fmt) de beeldschermindeling definiëren. Als er geen indelingsbestand is geactiveerd met SET FORMAT, wordt de beeldschermindeling van PFS:FILE gedefinieerd op basis van het standaardscherm van APPEND of EDIT.

Met de optie FW2 resulteert EXPORT in een database-frame voor Framework II.

Als het database-bestand van dBASE IV eerder werd geïmporteerd uit PFS:FILE, is er een bijbehorend indelingsbestand (.fmt).

Als het doelbestand al bestaat en SET SAFETY op ON is ingesteld, wordt u gewaarschuwd voordat het bestand eventueel wordt overschreven. Als SET SAFETY op OFF is ingesteld, wordt het bestand zonder meer overschreven.

OPMERKING
Met dBASE IV kunt u bestanden maken met velden die te groot zijn voor andere programma's. Hoewel deze velden in hun geheel worden geëxporteerd, is het mogelijk dat ze door andere programma's worden afgekapt. Controleer de beperkingen van andere programma's voordat u bestanden maakt met EXPORT.

Wanneer u bijvoorbeeld een indelingsbestand exporteert naar PFS:FILE, controleer dan of er niet meer dan 200 @...SAY...GET-commando's in staan. Tevens mag het formulier niet meer dan 21 rijen bevatten. De rijnummers in het formulier mogen dus lopen van 0 tot en met 20. Als een bestand niet aan deze eisen voldoet, kan PFS:FILE het bestand niet lezen.

Zie ook

COPY, IMPORT, SET FORMAT, SET SAFETY

FIND

Met het commando FIND zoekt u in een geïndexeerd database-bestand het eerste record met een indexsleutel die overeenkomt met de opgegeven tekenreeks of het opgegeven nummer. De database wordt zeer snel doorzocht.

Syntaxis

FIND <vaste tekens>

Gebruik

Met dit commando plaatst u de recordwijzer op het eerste record in een geïndexeerd database-bestand waarin de opgegeven vaste tekens voorkomen.

FIND en SEEK maken beide gebruik van een index, een indexbestand (.ndx) of een meervoudig indexbestand (.mdx), om snel gegevens op te zoeken in een database-bestand. De gebruikte index wordt de *hoofdindex* genoemd en kan worden geactiveerd met het commando SET INDEX, met het commando SET ORDER of met de INDEX- of ORDER-clausule van het commando USE. Met SEEK kunt u zoeken naar een uitdrukking maar met FIND kan dat niet.

LOCATE heeft ongeveer dezelfde functie als FIND en SEEK, maar doorzoekt het bestand sequentieel (record voor record). Het bestand hoeft dus niet te zijn geïndexeerd. LOCATE werkt beduidend langzamer.

Omdat FIND niet, zoals SEEK, uitdrukkingen in de commandoregel kan toetsen, moet u een tekengeheugenvariabele in combinatie met de macrosubstitutiefunctie & opgeven om naar de inhoud van de variabele te zoeken:

```
. FIND &Mgehvar
```

U kunt alleen met subtekenreeksen of gedeeltelijke zoeksleutels zoeken, als de zoekuitdrukking overeenkomt met de indexsleutel, te beginnen met het eerste teken, en als SET EXACT ingesteld is op OFF. U vindt met FIND geen subtekenreeksen van de zoeksleutel als SET EXACT is ingesteld op ON, omdat dan de gehele zoeksleutel wordt gezocht. Met FIND Smi vindt u bijvoorbeeld "Smit" als SET EXACT op OFF is ingesteld, maar niet als SET EXACT op ON is ingesteld.

FIND wordt beïnvloed door de instelling van SET DELETED. Als SET DELETED op ON is ingesteld, wordt de recordwijzer niet op gewiste records geplaatst. Ook de records die zijn uitgesloten door het commando SET FILTER, worden door FIND genegeerd.

Recordwijzer

Als de zoeksleutel is gevonden, wordt de recordwijzer op het record geplaatst waarin de sleutel is gevonden.

SET NEAR bepaalt waar de recordwijzer wordt geplaatst nadat het commando FIND is gegeven. Als SET NEAR op ON is ingesteld (of als NEAR = ON in het bestand Config.db staat) en er geen record wordt gevonden dat aan de zoeksleutel voldoet, wordt de recordwijzer op het volgende geïndexeerde record van het bestand geplaatst, vlak na de plaats waar de zoeksleutel zich volgens de index had kunnen bevinden. De functie FOUND() resulteert nog steeds in onwaar (.F.), omdat de sleutel niet is gevonden. De functie EOF() resulteert echter niet in waar (.T.), omdat de recordwijzer op een bijna overeenkomstig record in het bestand staat.

Als SET NEAR op OFF is ingesteld (de standaardinstelling) en de opgegeven tekenreeks of het opgegeven getal niet is gevonden, verschijnt de melding **Niet gevonden** op het scherm. U kunt deze melding voorkomen door SET TALK in te stellen op OFF. De recordwijzer gaat dan naar het einde van het bestand, FOUND() resulteert in onwaar (.F.) en EOF() in waar (.T.).

Als een ander bestand met SET RELATION aan het actieve bestand is gerelateerd en het commando FIND niets oplevert, staat de recordwijzer van het gerelateerde bestand altijd aan het einde van het bestand, ongeacht de instelling van NEAR.

De functie FOUND() resulteert alleen in waar (.T.) als de zoeksleutel werkelijk is gevonden, ongeacht de instelling van SET NEAR. De functie EOF() resulteert in waar (.T.) als SET NEAR op OFF is ingesteld en de sleutel niet is gevonden. Als SET NEAR op ON is ingesteld, resulteert EOF() alleen in waar als de waarde van de zoeksleutel hoger is dan alle sleutels in de index.

Richtlijnen voor het programmeren

Omdat het commando SEEK uitdrukkingen accepteert, wordt dat commando gewoonlijk gebruikt in programmabestanden waarin de uitdrukkingen door andere commando's en functies worden samengesteld en aan SEEK worden doorgegeven. FIND wordt meestal achteraf gebruikt vanaf de commandostip, hoewel FIND ook kan worden toegepast in programmabestanden.

Speciaal geval

FIND negeert voorloopspaties bij het zoeken naar een sleutel. De volgende twee commando's zijn identiek:

```
. FIND A
. FIND          A
```

Wanneer u een tekenreeks zoekt waarin voorloopspaties staan, plaatst u de zoeksleutel tussen enkele aanhalingstekens, dubbele aanhalingstekens of rechte haken. Bovendien moet u het exacte aantal voorloopspaties in de tekenreeks opnemen:

```
. FIND "        A"
```

Als er voorloopspaties in een geheugenvariabele staan, plaatst u de macrosubstitutiefunctie en de variabele tussen aanhalingstekens:

```
"&<Mgehvar>"
```

Als u zoekt naar een tekenreeks die begint met een scheidingsteken van dBASE, moet u de gehele tekenreeks afbakenen met een ander scheidingsteken:

```
. FIND ["Yamada"]
```

Voorbeelden

In deze voorbeelden wordt gebruik gemaakt van het database-bestand Afnemers, dat is geïndexeerd op basis van de uitdrukking Achternaam+Voornaam. Op de volgende wijze zoekt u het eerste record waarvan Achternaam begint met een hoofdletter M:

```
. USE Afnemers INDEX KIntnaam
Hoofdindex: KLNTNAAM
. FIND M
. ? Achternaam
Martinez
. SET EXACT ON
. FIND M
Niet gevonden
. SET EXACT OFF
```

U kunt ook zoeken naar een record met de naam Paterson (een dergelijk record bestaat niet):

```
. FIND Paterson
Niet gevonden
. ? EOF()
.T.
. SET NEAR ON
. FIND Paterson
Niet gevonden
. ? EOF()
.F.
. ? FOUND()
.F.
. ? Achternaam
Peters
```

Zie ook

EOF(), FOUND(), INDEX, KEY(), LOCATE, LOOKUP(), MDX(), NDX(), SEEK, SEEK(), SET DELETED, SET FILTER, SET INDEX, SET NEAR, SET ORDER, SET RELATION, TAG(), USE

FUNCTION

Het commando FUNCTION wordt gebruikt om een door de gebruiker gedefinieerde functie (UDF) te definiëren.

Syntaxis

```
FUNCTION <UDF-naam>
        [PARAMETERS]
        .
        <andere dBASE IV-commando's>
        .
RETURN <uitdrukking>
```

Gebruik

De UDF is een speciaal soort procedure die in een dBASE IV-uitdrukking kan worden gebruikt en een waarde oplevert. Het definiëren van een UDF bestaat uit het commando FUNCTION gevolgd door een naam die u aan een bepaalde procedure verbindt. Een UDF-naam mag maximaal acht tekens lang zijn, moet beginnen met een letter en kan spaties bevatten.

De UDF-naam kan optioneel worden gevolgd door een lijst parameters, dBASE IV-commando's of programmastructuren. De procedure wordt afgesloten met het commando RETURN. Bij RETURN wordt de waarde opgegeven die de UDF moet opleveren. Afhankelijk van de complexiteit van het programma kunt u maximaal 32 UDF's nesten in de UDF waarmee het programma wordt aangeroepen. In de praktijk is dit aantal echter meestal kleiner.

Met het optionele commando PARAMETERS worden namen van parameters ingevoerd die door de UDF als argument worden gebruikt. Als u parameters aan de UDF wilt toevoegen, staat de plaats van het commando PARAMETERS vast: het moet het eerste commando zijn in de definitie van de UDF. De parameters zijn voor de UDF lokaal. De volgorde van de lijst met parameters is ook de volgorde waarin parameters aan de UDF moeten worden toegekend.

Door het commando RETURN wordt <uitdrukking> teruggezonden die bij de definitie van de functie is opgegeven.

Als in dBASE IV de naam van een UDF in een commandoregel verschijnt, wordt naar de betreffende UDF gezocht, waarna deze wordt uitgevoerd en een enkele waarde wordt teruggezonden. Na beëindiging van de UDF berust de besturing van het programma weer bij het commando van waaruit de UDF is aangeroepen.

WAARSCHUWING
Gebruik voor uw UDF niet de naam van een bestaande dBASE IV-functie. In dBASE IV heeft een bestaande functie hogere prioriteit dan een UDF en dus zal de bestaande functie worden uitgevoerd en niet uw UDF.

Onderbrekingen verwerken

De uitvoering van een dBASE IV-commando kan worden onderbroken voor de uitvoering van een ander dBASE IV-commando, een UDF of een ander programma. De meeste programma's van dBASE IV kunnen op meerdere niveaus van het programma worden onderbroken door een UDF of door een ON-commmando.

Het is niet toegestaan met een onderbrekende UDF de structuur van een actief database-bestand te sluiten of te wijzigen. Evenmin mag door de onderbrekende UDF een actief venster uit het geheugen worden verwijderd. U mag bijvoorbeeld niet vanuit de modus BROWSE een UDF aanroepen die probeert met het commando USE een andere database in hetzelfde werkgebied te openen. In zo'n geval krijgt u een foutmelding. dBASE IV beschermt het werkgebied en elk geactiveerd venster, zodat u na de onderbreking hiernaar terug kunt keren.

In dBASE IV zijn bepaalde wijzigingen zonder beperking toegestaan. De wijzigingen worden doorgegeven aan het onderbroken commando. Zo kan een commando ON KEY een UDF aanroepen die een waarde in een BROWSE-veld wijzigt. De gewijzigde waarde wordt doorgegeven aan BROWSE en het veld wordt gewijzigd.

Alle commando's die een database-bestand zouden sluiten of de structuur ervan wijzigen, worden geblokkeerd als DBTRAP op ON is ingesteld, zodat het oorspronkelijke bestand bewaard blijft.

Tot de geblokkeerde commando's behoren USE, CLOSE DATA, CONVERT, PACK, MODIFY STRUCTURE en ZAP.

U kunt vensters niet uit het geheugen verwijderen, zodat het oorspronkelijke venster behouden blijft. Tot de geblokkeerde commando's behoren RELEASE WINDOWS, CLEAR WINDOW, DEFINE WINDOW <naam actieve venster> en RESTORE WINDOW <naam actieve venster>.

U kunt een venster wel verplaatsen of de kleurinstelling ervan wijzigen door in het venster respectievelijk de commando's MOVE en SET COLOR te geven.

Na de onderbreking wordt het oorspronkelijk actieve venster weer hersteld. Als u tijdens de onderbrekende UDF de positie of de kleurinstelling van een venster hebt gewijzigd, wordt het venster na afloop op de nieuwe positie en volgens de nieuwe kleurinstelling weergegeven.

Er zijn veel redenen waarom onderbrekingen kunnen plaatsvinden. Hierna vindt u een aantal manieren waarop een commando BROWSE door een UDF kan worden onderbroken.

- LOCK <Nuitdr>
- WIDTH <Nuitdr>
- SET FIELDS <naam rekenveld> = <uitdr>
- SET RELATION TO <uitdr>
- SET FILTER TO <Luitdr>
- @...RANGE <lage uitdr>, <hoge uitdr>
- @...VALID <Luitdr>

- @...WHEN <Luitdr>
- @...DEFAULT <uitdr>
- @...MESSAGE <Tuitdr>

De uitdrukking WIDTH wordt direct getoetst. De uitdrukkingen @...RANGE en @...VALID worden alleen uitgevoerd als u een wijziging aanbrengt in het veld of als u het sleutelwoord REQUIRED in de syntaxis opneemt.

WAARSCHUWING
Zorg ervoor dat door de UDF de verwerkingsvoorwaarden van het onderbroken commando niet worden gewijzigd. Het betreft onder meer de relatie, de veldenlijst, het filter en het bereik van de actieve database. Wanneer u de vermelde instellingen wijzigt, kan dat onvoorspelbare resultaten opleveren als u na beëindigen van de UDF terugkeert.

Niet-recursieve commando's

Een aantal nauw verwante commando's mogen elkaar in hetzelfde werkgebied niet aanroepen. Dit geldt ongeacht de instelling van DBTRAP. De commando's zijn:

BROWSE, EDIT, CHANGE, INSERT en APPEND.

Twee andere commando's kunnen elkaar zelfs in verschillende werkgebieden niet aanroepen. Dit zijn de commando's:

LIST en DISPLAY.

Bovendien kunnen ontwerpschermen van het Control Center en de commando's ASSIST en CREATE/MODIFY niet recursief worden gebruikt.

Indexeren met behulp van UDF's

Als DBTRAP op ON is ingesteld, is de indexering beveiligd tegen onderbrekingen. Om een index te maken die een UDF als sleuteluitdrukking gebruikt, moet DBTRAP op OFF worden ingesteld en de volgende syntaxis worden gebruikt.

INDEX ON <UDF-naam>...[FOR <UDF-naam>]

Wanneer u indexeert en een UDF in de sleuteluitdrukking gebruikt, kan dit een ongeldige index opleveren. Tijdens het indexeren worden geen foutmeldingen weergegeven als de UDF een niet te indexeren waarde als uitkomst heeft.

Als de indexering correct verloopt, maar de index de UDF op een later tijdstip niet kan vinden, wordt een standaardwaarde in de index opgenomen, waardoor die ongeldig wordt. Het is belangrijk op de hoogte te zijn van beide problemen die zich bij het gebruik van UDF's in de indexsleutel kunnen voordoen.

Uitzonderingen bij SQL

De definitie van een UDF kan worden opgenomen in een .prs-bestand, terwijl een SQL-programma een UDF kan aanroepen met een dBASE IV-commando dat wordt gebruikt door het .prs-bestand.

De syntaxis voor SQL-taal stelt enkele beperkingen aan het gebruik van UDF's:

- U mag een UDF niet gebruiken in een SQL-opdracht
- U mag SQL-opdrachten niet gebruiken in een UDF
- U mag in een UDF die in een .prs-bestand staat, geen dBASE IV-commando's gebruiken die in de SQL-modus niet toegestaan zijn.

Raadpleeg bijlage H voor een overzicht van de commando's die in de SQL-modus zijn toegestaan.

& Macrosubstitutie

In UDF's is & macrosubstitutie toegestaan.

Voorbeelden

De volgende UDF (Plus_BTW) geeft de uiteindelijke kosten van een produkt door de BTW op te tellen bij de prijs. Deze door de gebruiker gedefinieerde functie vereist dat de prijs en de BTW als parameters moeten worden verwerkt.

```
FUNCTION Plus_BTW
PARAMETERS M_prijs, M_belast
M_kosten = (M_prijs * M_belast) + M_prijs
RETURN (M_kosten)
* EOF Plus_BTW.prg
```

Het bestand moet in de actieve bestandsindex staan:

```
. SET TALK OFF
. M_belast = ,10
. ? Plus_BTW(100, M_belast)
                                    110
```

Zie ook

DO, PARAMETERS, PROCEDURE, RETURN, SET DBTRAP, SET PROCEDURE

Hoofdstuk 1, "Basisbegrippen"

GO/GOTO

Met het commando GO/GOTO plaatst u de recordwijzer op een bepaald record in het actieve database-bestand.

Syntaxis

GO/GOTO BOTTOM/TOP [IN <alias>]

of

GO/GOTO [RECORD] <recordnummer> [IN <alias>]

of

<recordnummer> [IN <alias>]

Gebruik

Als er geen index actief is, verwijzen TOP en BOTTOM naar het eerste respectievelijk laatste record in het database-bestand. Is er wel een index actief, dan verwijzen TOP en BOTTOM naar het eerste en laatste record in het indexbestand. GO <recordnummer> verwijst naar het opgegeven recordnummer, niet naar een positie in het indexbestand. U kunt de recordwijzer ook op een bepaald record plaatsen door alleen de numerieke uitdrukking op te geven, zonder het commando GO/GOTO.

Als SET DELETED op ON is ingesteld, kunt u de recordwijzer op een record met een wismarkering plaatsen door het recordnummer op te geven. U kunt de recordwijzer dan ook op records plaatsen die worden weggelaten door SET FILTER. Met het commando EDIT krijgt u echter geen toegang tot deze records.

Als verschillende bestanden met elkaar zijn gerelateerd en de recordwijzer met GOTO wordt verplaatst in het ouderbestand, wordt de recordwijzer in het kindbestand op een gerelateerd record geplaatst. Als er geen gerelateerd record is, wordt de recordwijzer van het kindbestand aan het einde van het bestand geplaatst. EOF() resulteert dan in waar (.T.). Als u daarentegen de recordwijzer in het kindbestand verplaatst, blijft de recordwijzer in het ouderbestand op dezelfde positie.

Opties

U kunt de recordwijzer in een ander werkgebied plaatsen met de clausule IN. Met deze clausule kunt u het database-bestand in een ander werkgebied bewerken zonder het bestand met SELECT te kiezen als het actieve werkgebied. De <alias> mag een van de volgende waarden zijn:

- een getal uit de reeks 1 tot en met 10
- een letter uit de reeks A tot en met J
- een aliasnaam, hetzij standaard, hetzij opgegeven met de optie ALIAS van het commando USE

- een numerieke uitdrukking die een getal uit de reeks 1 tot en met 10 oplevert (tussen haakjes als het een geheugenvariabele is)
- een tekenuitdrukking die een letter uit de reeks A tot en met J of een aliasnaam oplevert

In het gedeelte "Bestandsnamen en aliassen" van hoofdstuk 1 vindt u meer informatie over het gebruik van bestandsnamen en aliassen.

Richtlijnen voor het programmeren

In een netwerk verschijnt de melding **Relatierecord is in gebruik door een andere gebruiker** als het gerelateerde record is vergrendeld en u probeert de inhoud van dat record te wijzigen. U zult fout 142 vinden met een ON ERROR-routine en u kunt opnieuw proberen het record te vergrendelen.

Voorbeelden

In de volgende voorbeelden wordt gebruik gemaakt van het database-bestand Afnemers.

Op de volgende wijze plaatst u de recordwijzer op record vijf van het database-bestand:

```
. USE Afnemers
. 5
AFNEMERS: Recordnr.          5
```

Sla het recordnummer op met behulp van een geheugenvariabele:

```
. M_recnr = RECNO()
                    5,00
. GO TOP
AFNEMERS: Recordnr.          1
. GO M_recnr
AFNEMERS: Recordnr.          5
```

Zie ook

ON ERROR, RECNO(), SELECT, SET DELETED, SET FILTER, SET RELATION, SKIP

HELP

HELP is een menugestuurd commando dat informatie geeft over dBASE IV.

Syntaxis

HELP [<dBASE IV-sleutelwoord>]

Gebruik

Het commando HELP maakt gebruik van de bestanden Dbase1.hlp en Dbase2.hlp van dBASE IV.

Het sleutelwoord moet een commando, een functie of de naam van een hulpscherm van dBASE IV zijn. In plaats van HELP te typen kunt u ook op **F1 Hulp** drukken om de inhoudsopgave van het hulpsysteem op te vragen. Daarin staan de belangrijkste onderwerpen vermeld. Met de pijltoetsen verplaatst u de selectiebalk door het menu. Druk op ↵ om de gemarkeerde optie te kiezen.

U kunt ook teruggaan en eerdere schermen opnieuw lezen, een scherm afdrukken, aanverwante onderwerpen bekijken of alleen de syntaxis en het voorbeeld van een commando bekijken.

U verlaat het hulpsysteem door op **Esc** te drukken. De hulptekst blijft op het scherm staan, zodat u de informatie kunt gebruiken terwijl u bezig bent op de commandoregel. De tekst verdwijnt als u op ↵ drukt.

Voorbeeld

Op de volgende wijze vraagt u informatie over het commando RUN:

```
. HELP RUN
```

Zie ook

SET HELP

IF is een commando voor gestructureerd programmeren, waarmee u voorwaarden kunt verbinden aan de uitvoering van commando's. De IF-structuur wordt beëindigd met het commando ENDIF.

Syntaxis

```
IF <voorwaarde>
   <commando's>
[ELSE
   <commando's>]
ENDIF
```

Gebruik

IF kan alleen worden gebruikt in programma's, niet vanaf de commandostip.

De commando's binnen een IF...ENDIF-structuur moeten op de juiste wijze worden genest. Ook het commando IF zelf mag worden genest.

Met IF <voorwaarde> wordt een voorwaarde gesteld of een logische uitdrukking opgegeven, bijvoorbeeld A = B of Numvar < 11. De voorwaarde of logische uitdrukking wordt getoetst, en als dit resulteert in de logische waarde waar (.T.), worden alle eropvolgende commando's uitgevoerd tot de eerstvolgende opdracht ELSE (of ENDIF als er geen opdracht ELSE is opgegeven). Vervolgens wordt het eerste commando na ENDIF uitgevoerd.

Als de voorwaarde resulteert in de logische waarde onwaar (.F.), wordt het programma voortgezet vanaf ELSE of ENDIF, afhankelijk van welke opdracht het eerst komt. De commando's tussen ELSE en ENDIF worden uitgevoerd als de voorwaarde onwaar is.

Wanneer het commando IF meermalen in een commandostructuur voorkomt, verwijst ELSE naar de laatst uitgevoerde IF in de geneste structuur.

De ruimte na ENDIF op de commandoregel kan worden gebruikt voor commentaar. De commentaaraanduiding && is niet noodzakelijk, al geeft de compiler dan wel een waarschuwing.

Voorbeeld

Vergelijk dit voorbeeld met het voorbeeld dat is gegeven bij het commando DO CASE. De volgende geneste IF-structuur bepaalt de grootte van een geheugenvariabele en geeft de bijbehorende melding weer:

```
IF M_waarde > 100
   ? "Waarde is groter dan 100."
ELSE
   IF M_waarde > 10
      ? "Waarde is groter dan 10."
   ELSE
      IF M_waarde > 1
         ? "Waarde is groter dan 1."
      ELSE
         ? "Waarde is 1 of minder."
      ENDIF
   ENDIF
ENDIF
```

Zie ook

DO CASE, DO WHILE, IIF(), SCAN

IMPORT

Met het commando IMPORT zet u formulieren van PFS:FILE, database-bestanden van dBASE II, database- en spreadsheetframes van Framework II, gegevensbestanden van RapidFile en .WK1-spreadsheets van Lotus om in dBASE IV-bestanden.

Syntaxis

IMPORT FROM <bestandsnaam> [TYPE] PFS/DBASEII/FW2/RPD/WK1

Standaardinstelling

De bestandsnaam moet worden voorzien van een extensie wanneer deze afwijkt van de standaardextensie. Gewoonlijk hebben dBASE II-bestanden de extensie .dbf, Framework II-bestanden de extensie .fw2, RapidFile-bestanden de extensie .rpd en Lotus 1-2-3 bestanden (versie 2.x) de extensie .wk1. PFS:FILE-formulieren hebben meestal geen extensie. De bestanden waarnaar de gegevens door dBASE IV worden geconverteerd, krijgen meestal dezelfde naam als het originele bestand, maar met de extensie .dbf. Wanneer u bestanden van dBASE II importeert, moet de extensie van het bestand worden gewijzigd in .db2 voordat het bestand wordt geïmporteerd.

Wanneer een catalogus geopend is, wordt het nieuwe bestand aan de catalogus toegevoegd. De records in het nieuwe database-bestand mogen niet groter zijn dan 4000 bytes.

Met IMPORT maakt u het uitvoerbestand in hetzelfde station of in dezelfde bestandsindex als het oorspronkelijke bestand en neemt u het in gebruik. Daarna worden de geïmporteerde gegevens naar het nieuwe uitvoerbestand weggeschreven. Met DISPLAY STATUS en DISPLAY STRUCTURE kunt u gegevens over het geïmporteerde bestand op het scherm weergeven.

Gebruik

Om er zeker van te zijn dat PFS-bestanden goed worden geïmporteerd, moet u controleren of het bestand aan de volgende voorwaarden voldoet:

- 255 gegevens per formulier. Een record kan uit maximaal 255 velden bestaan. Houd er rekening mee dat kopregels en commentaar in het PFS:FILE-formulier ook in velden worden omgezet.
- 254 tekens per gegevens-onderdeel. 254 is de maximale lengte van een tekenveld in dBASE IV.

Wanneer u een PFS:FILE-formulier importeert, worden er drie bestanden gemaakt: een database-bestand (.dbf), een indelingsbestand (.fmt) en een gecompileerd indelingsbestand (.fmo). De drie bestanden hebben dezelfde naam en krijgen de standaardextensie.

Zie ook

APPEND FROM, COPY, DISPLAY STATUS, DISPLAY STRUCTURE, EXPORT, SET FORMAT, USE

INDEX

Met het commando INDEX maakt u een index waarin de records van een database-bestand in alfabetische, chronologische of numerieke volgorde worden gerangschikt.

Syntaxis

INDEX ON <sleuteluitdrukking>
    TO <.ndx-bestand> / TAG <labelnaam>[OF <.mdx-bestand>]
        [FOR<voorwaarde>] [UNIQUE][DESCENDING]

Standaardinstelling

Tenzij u anders bepaalt, wordt het indexbestand op het standaardstation en in de standaardbestandsindex opgeslagen. Als u geen extensie opgeeft, is de standaardextensie .ndx of .mdx van toepassing.

Wanneer u alleen INDEX ↵ typt (zonder sleutelwoorden of opties), wordt u gevraagd om de sleuteluitdrukking, die gewoonlijk wordt opgegeven met de clausule ON, en om de naam van het doelbestand, die gewoonlijk wordt opgegeven met de clausule TO.

Wanneer SET SAFETY op ON is ingesteld (de standaardinstelling), wordt u gewaarschuwd voordat een indexbestand (.ndx) of een label met dezelfde naam wordt overschreven.

Als u wel TAG kiest, maar geen .mdx-bestand opgeeft met de clausule OF, wordt de label in het bijbehorende .mdx-bestand geplaatst. Kiest u geen TAG, dan wordt een .ndx-bestand gemaakt.

Wanneer de FOR-clausule of de UNIQUE-optie niet wordt opgegeven, bevat de index alle records. De FOR-clausule kan alleen bij .mdx-labels worden gebruikt. UNIQUE kan bij .ndx-bestanden en .mdx-labels worden gebruikt.

De records worden in oplopende volgorde geïndexeerd, tenzij de optie DESCENDING is opgegeven. Deze optie is alleen toegestaan in .mdx-labels.

Gebruik

In geïndexeerde database-bestanden kunt u met de commando's FIND en SEEK, of met de functies LOOKUP() en SEEK(), de recordwijzer direct op het eerste record plaatsen waarvan de gegevens overeenkomen met de opgegeven uitdrukking. De actieve index bepaalt hoe de recordwijzer in het database-bestand wordt verplaatst.

De index wordt als indexbestand of als label in een meervoudig indexbestand opgeslagen, en bevat de sleutelwaarden en de daarmee overeenkomende nummers van de records in het database-bestand. De feitelijke volgorde van de records in het originele database-bestand wordt door het commando INDEX niet gewijzigd.

Een samengesteld indexbestand kan maximaal 47 labels bevatten. Elke label kan het database-bestand ordenen volgens een index. Voor de namen van labels gelden dezelfde regels als voor de namen van variabelen. De naam mag niet langer zijn dan tien tekens, moet beginnen met een letter en mag bestaan uit letters, cijfers en onderstrepingstekens. Namen van indexbestanden (zowel .ndx als .mdx) moeten aan dezelfde voorwaarden voldoen als alle andere bestandsnamen: ze mogen maximaal acht tekens lang zijn en ze mogen alleen uit tekens bestaan die ook door het besturingssysteem worden toegestaan.

Een bijbehorend .mdx-bestand (een "produktiebestand") is een .mdx-bestand dat automatisch wordt geopend wanneer een database-bestand in gebruik wordt genomen. Elk database-bestand kan één .mdx-bestand hebben. Het meervoudig indexbestand heeft dezelfde naam als het database-bestand, maar heeft de extensie .mdx in plaats van de extensie .dbf. In de aanhef van het database-bestand wordt met een vlag aangegeven of er een bijbehorend .mdx-bestand bestaat.

Wanneer u een .ndx-bestand wilt maken, geeft u TO <.ndx-bestand> op. U voegt een label toe aan het bijbehorende .mdx-bestand door TAG <labelnaam> op te geven. Als u de label in een ander .mdx-bestand dan het bijbehorende bestand wilt plaatsen, geeft u TAG <labelnaam> OF <.mdx-bestand> op.

Wanneer u een index maakt, wordt dat de actieve index en staan de records in de nieuwe indexvolgorde. Om een andere index te kiezen, geeft u het commando SET INDEX of SET ORDER. Andere actieve indexen hebben geen invloed op de recordwijzer en zijn alleen geopend om te kunnen worden bijgewerkt wanneer er record-gegevens worden gewijzigd, toegevoegd of verwijderd. Wanneer een database-bestand wordt gewijzigd en dit gevolgen heeft voor de indexsleutel, moet het desbetreffende indexbestand namelijk geopend zijn, anders wordt de index niet bijgewerkt. U kunt een index alsnog bijwerken door het indexbestand te openen en het commando REINDEX te geven.

Wanneer u een oplopende index maakt, moet de sleuteluitdrukking een veld zijn of een geldige uitdrukking. De sleuteluitdrukking mag niet langer zijn dan 220 tekens. De sleutel zelf, het resultaat van de uitgevoerde sleuteluitdrukking, mag niet groter zijn dan 100 tekens.

Het gegevenstype van de indexsleutel bepaalt of de records in chronologische (datumuitdrukkingen), numerieke (numerieke uitdrukkingen) of in ASCII-volgorde (tekenuitdrukkingen) worden gerangschikt. Als de sleuteluitdrukking meerdere velden omvat, moeten deze allemaal van hetzelfde gegevenstype zijn. Met behulp van dBASE-functies kunt u velden omzetten in het juiste type. Enkele van de meest gebruikte functies voor sleuteluitdrukkingen zijn STR(), SUBSTR(), CTOD(), DTOC(), DTOS(), YEAR(), MONTH(), DAY() en VAL().

Opties

Met FOR <voorwaarde> worden alleen die records geïndexeerd die aan een opgegeven voorwaarde voldoen. FOR kan alleen bij .mdx-labels worden gebruikt. Als er geen records zijn die aan de FOR-voorwaarde voldoen, wordt er toch een lege label gemaakt en onderhouden.

De FOR-voorwaarde kan geen rekenvelden bevatten. De voorwaarde mag maximaal 220 tekens lang zijn (inclusief ingebedde spaties).

De optie UNIQUE heeft hetzelfde resultaat als het commando SET UNIQUE ON voor INDEX. Wanneer het sleutelveld in meerdere records dezelfde waarde bevat, wordt alleen het eerste record met die waarde in het .ndx-bestand opgenomen. Wanneer een indexbestand dat met de optie UNIQUE is gemaakt, met REINDEX opnieuw wordt geïndexeerd, behoudt het bestand zijn UNIQUE-status, ongeacht of SET UNIQUE op ON of OFF is ingesteld.

Een UNIQUE-index wordt slechts eenmaal verwerkt door dBASE IV. Een sleutelwaarde die eerder is verborgen, wordt niet automatisch bijgewerkt wanneer deze wordt gewijzigd. Alle sleutelwaarden in een UNIQUE-index worden door REINDEX expliciet bijgewerkt.

Met DESCENDING maakt u een .mdx-label in aflopende numerieke volgorde. De optie DESCENDING kan alleen verwijzen naar de gehele indexuitdrukking en niet naar slechts één element in de uitdrukking, zoals een veld. U kunt geen .ndx-bestand maken met de optie DESCENDING.

Tips

U kunt sleuteluitdrukkingen opgeven waarmee het veldtype wordt gewijzigd. Wanneer u bijvoorbeeld een datumveld chronologisch wilt indexeren en u voor iedere datum de achternamen alfabetisch wilt indexeren, kunt u de volgende uitdrukking opgeven:

```
INDEX ON DTOS(<datum>) + Achternaam TO <indexbestand>
```

Wanneer de sleuteluitdrukking wordt uitgevoerd, moet dat een sleutelwaarde met een vaste lengte opleveren. U kunt wel een index maken met sleutelwaarden van variabele lengte, maar de index is dan niet betrouwbaar. Let er bij het maken van een index altijd op dat u een vaste lengte opgeeft bij functies die daar een parameter voor hebben, zoals STR() en SUBSTR().

Bewerkingen die een database-bestand sequentieel doorlopen, worden meestal langzamer uitgevoerd wanneer er een index is geopend. Geef het commando SET ORDER wanneer u de index niet gebruikt voor de plaatsing van de recordwijzer, maar de indexsleutels wel moeten worden bijgewerkt. Door SET ORDER zijn de gegevens sneller toegankelijk wanneer er een indexbestand is geopend.

Met LIST/DISPLAY STATUS wordt de FOR<voorwaarde> voor de indexen weergegeven (als er ten minste een voorwaarde is opgegeven) na de regel waarop de <sleuteluitdrukking> zich bevindt.

Speciale gevallen

Als u TAG opgeeft, maar de naam van een .mdx-bestand weglaat, wordt in de aanhef van het database-bestand gecontroleerd of er een bijbehorend .mdx-bestand bestaat. Is dat het geval, dan wordt de label in dat bestand geplaatst. Als er geen bijbehorend .mdx-bestand is, wordt dat alsnog gemaakt. Daarbij is het mogelijk dat dBASE IV toch een .mdx-bestand vindt met dezelfde naam als het .mdx-bestand dat dBASE IV probeert te maken. In dat geval wordt de aanhef van het database-bestand bijgewerkt zodat dit bestand het bijbehorende .mdx-bestand wordt. Er wordt dan dus geen nieuw bestand gemaakt.

Als u de clausule OF opgeeft en het .mdx-bestand niet geopend is, wordt het .mdx-bestand

geopend voordat de index wordt gemaakt. Als het opgegeven .mdx-bestand niet wordt gevonden, wordt er een nieuw .mdx-bestand gemaakt met de bestandsnaam die in de OF-clausule wordt opgegeven.

INDEX wordt niet beïnvloed door filters die zijn ingesteld met SET FILTER of SET DELETED. Alle records in het database-bestand worden opgenomen in de index, tenzij u FOR <voorwaarde> opneemt of de optie UNIQUE gebruikt om dubbele records te voorkomen.

U kunt ook indexeren op basis van geheugenvariabelen en velden in andere geopende database-bestanden. Indexen op basis van gegevens die niet in het actieve database-bestand staan, worden echter niet altijd bijgewerkt wanneer de gegevens worden gewijzigd. Als u een index opent die verwijst naar een niet-bestaande geheugenvariabele, veroorzaakt dat een fout.

Als u een .ndx- of .mdx-bestand maakt terwijl er een catalogus geopend is, wordt het indexbestand aan de catalogus toegevoegd. Indexlabels die in een .mdx-bestand staan, worden echter niet in de catalogus opgenomen.

In een netwerk kan een database-bestand alleen worden geïndexeerd als het in exclusief gebruik is. Is dat niet het geval, dan verschijnt de foutmelding **Bestand moet in modus EXCLUSIVE zijn geopend**.

U kunt het commando INDEX niet gebruiken in een actieve transactie wanneer:

- er op dat moment een indexbestand is geopend dat niet het bijbehorende indexbestand (het "produktiebestand") is
- er een bestaand .ndx- of .mdx-bestand wordt overschreven
- er een bestaande .mdx-label wordt overschreven

Voorbeelden

Op de volgende wijze indexeert u het database-bestand Transact op het veld Klant_nr:

```
. USE Transact
. INDEX ON Klant_nr TO Klant_nr
  100% geïndexeerd          12 records geïndexeerd
. LIST

Recordnr.   KLANT_NR   BESTEL_NR   DAT_TRANS   FACTUUR   TOTAAL_REK
        9   A00005     87-113      24-03-87    .T.          125,00
        1   A10025     87-105      03-02-87    .T.         1850,00
       12   A10025     87-116      10-04-87    .F.         1500,00
       10   B12000     87-114      30-03-87    .F.          450,00
        2   C00001     87-106      10-02-87    .T.         1200,00
                .
                .
                .
        7   L00002     87-111      11-03-87    .F.         1000,00
```

Op de volgende wijze indexeert u het bestand op cliëntnummer en op het totale bedrag van hun transacties:

```
. INDEX ON Klant_nr + STR(Totaal_rek,10,2) TO Op_bedr
  100% geïndexeerd          12 records geïndexeerd
. LIST

Recordnr.   KLANT_NR   BESTEL_NR   DAT_TRANS   FACTUUR   TOTAAL_REK
        9   A00005     87-113      24-03-87    .T.          125,00
       12   A10025     87-116      10-04-87    .F.         1500,00
        1   A10025     87-105      03-02-87    .T.         1850,00
       10   B12000     87-114      30-03-87    .F.          450,00
       11   C00001     87-115      01-04-87    .F.          165,00
                .
                .
                .
        7   L00002     87-111      11-03-87    .F.         1000,00
```

Voor een alfabetische lijst van alle cliëntnummers gebruikt u de optie UNIQUE:

```
. INDEX ON Klant_nr TO Cliënten UNIQUE
  100% geïndexeerd          7 records geïndexeerd
. LIST Klant_nr

Recordnr.   Klant_nr
        9   A00005
        1   A10025
       10   B12000
        2   C00001
        3   C00002
        5   L00001
        7   L00002
```

Als u een index wilt maken op basis van de voorwaarde dat alleen records worden opgenomen waarvan het veld Totaal_rek ƒ 1000,00 of meer bevat, gaat u als volgt te werk:

```
. INDEX ON Klant_nr TAG Groot FOR Totaal_rek <= 1000
  100% geïndexeerd          6 records geïndexeerd
. LIST

Recordnr.   KLANT_NR   BESTEL_NR   DAT_TRANS   FACTUUR   TOTAAL_REK
        1   A10025     87-105      03-02-87    .T.         1850,00
       12   A10025     87-116      10-04-87    .F.         1500,00
        2   C00001     87-106      10-02-87    .T.         1200,00
        4   C00001     87-108      23-02-87    .T.         1250,00
        3   C00002     87-107      12-02-87    .T.         1250,00
        7   L00002     87-111      11-03-87    .F.         1000,00
```

Op de volgende wijze maakt u een indexlabel van Transact in omgekeerde chronologische volgorde:

```
. INDEX ON Dat_trans TAG Recent DESCENDING
  100% geïndexeerd          12 records geïndexeerd
. LIST

Recordnr.   KLANT_NR   BESTEL_NR   DAT_TRANS   FACTUUR   TOTAAL_REK
       12   A10025     87-116      10-04-87    .F.          1500,00
       11   C00001     87-115      01-04-87    .F.           165,00
       10   B12000     87-114      30-03-87    .F.           450,00
        9   A00005     87-113      24-03-87    .T.           125,00
        8   L00001     87-112      20-03-87    .T.           700,00
              .
              .
              .
        1   A10025     87-105      03-02-87    .T.          1850,00
```

Zie ook

BEGIN/END TRANSACTION, CLOSE, COPY INDEX, COPY TAG, CTOD(), DAY(), DELETE TAG, DTOC(), DTOS(), FIND, KEY(), LIST STATUS, LOOKUP(), MDX(), MONTH(), NDX(), ORDER(), REINDEX, SEEK, SEEK(), SET DELETED, SET EXCLUSIVE, SET FILTER, SET INDEX, SET NEAR, SET ORDER, SET SAFETY, SET UNIQUE, SORT, STR(), SUBSTR(), TAG(), USE, VAL(), YEAR()

Op de verschillen tussen INDEX en SORT wordt bij het commando SORT dieper ingegaan.

INPUT

Het commando INPUT wordt vooral toegepast om de gebruiker te vragen een uitdrukking in te voeren vanaf het toetsenbord. De invoer van gegevens wordt beëindigd met ↵.

Syntaxis

INPUT [<aanwijzing>] TO <geheugenvariabele>

Gebruik

Met dit commando maakt u, indien nodig, een geheugenvariabele voor de uitdrukking die als antwoord wordt ingevoerd.

De aanwijzing moet een tekenuitdrukking zijn. Als de aanwijzing een vast teken in plaats van een geheugenvariabele is, moet de aanwijzing tussen enkele aanhalingstekens (‘ ‘), dubbele aanhalingstekens (“ “) of vierkante haken ([ ]) worden geplaatst.

U kunt elke in dBASE geldige uitdrukking invoeren als antwoord op het commando INPUT. Het type van de ingevoerde uitdrukking bepaalt het type van de geheugenvariabele die wordt gemaakt.

Wanneer u het antwoord hebt ingevoerd, drukt u op ↵. Als u op ↵ drukt zonder eerst een antwoord te geven, verschijnt de aanwijzing opnieuw.

Richtlijnen voor het programmeren

Wanneer het antwoord van het type teken moet zijn, geeft u het commando ACCEPT, WAIT of @...GET. In het geval van ACCEPT wordt verondersteld dat alle invoer van de gebruiker van het type teken is. De invoer hoeft niet te worden voorzien van scheidingstekens. In tegenstelling tot andere commando’s accepteert INPUT complexe uitdrukkingen.

Wanneer het antwoord van het type datum moet zijn, kunt u in de aanwijzing de instructie opnemen dat de datum tussen accolades moet worden ingevoerd. Accolades zijn de scheidingstekens voor datums. Wanneer de invoer van het type teken is, zet u de tekenvariabele met behulp van de functie CTOD() om in een datumvariabele.

Als er al een geheugenvariabele met de opgegeven naam bestaat, wordt deze overschreven, tenzij u de ene variabele PUBLIC en de andere PRIVATE maakt.

Zie ook

@, ACCEPT, PRIVATE, PUBLIC, READ, STORE, WAIT

INSERT

Met het commando INSERT voegt u op de positie van de recordwijzer een nieuw record toe aan het database-bestand.

Syntaxis

INSERT [BEFORE] [BLANK]

Gebruik

Het nieuwe record wordt schermgroot weergegeven, zodat u gegevens kunt invoeren. U kunt alleen in dit record gegevens invoeren. Het nieuwe record wordt direct volgend op het actieve record ingevoegd. Wanneer bijvoorbeeld het nummer van het actieve record 5 is, maakt u met het nieuwe record nummer 6. Het oude record 6 wordt record 7, enzovoort. Als u op **PgDn** of **PgUp** drukt, wordt het voorafgaande record geheel naar het zojuist ingevoegde, lege record gekopieerd.

In twee gevallen werkt INSERT als APPEND en worden één voor één meerdere records toegevoegd aan het einde van een bestand: als het bestand is geïndexeerd of als de recordwijzer al aan het einde van het bestand staat.

U kunt gegevens in een memoveld invoeren door de cursor op het veld te plaatsen en op **Ctrl-Home** te drukken. U verlaat het memoveld door op **Ctrl-End** (de wijzigingen worden opgeslagen) of op **Esc** (wijzigingen worden niet opgeslagen) te drukken. Bij het bewerken van het memoveld kunt u dezelfde stuurtoetsen gebruiken als bij MODIFY COMMAND. Met **F1** kunt u het menu met de te gebruiken toetsen in- en uitschakelen.

Met de pijltoetsen verplaatst u de cursor binnen een record. Met **Esc** beëindigt u het proces zonder een nieuw record in te voegen. Daarbij verschijnt de melding **Record niet ingevoegd**. Met **Ctrl-End** beëindigt u het proces, en wordt het record ingevoegd.

Opties

Wanneer u de clausule BEFORE opgeeft, wordt het nieuwe record vóór het actieve record ingevoegd in plaats van erna. Wanneer bijvoorbeeld het nummer van het actieve record 5 is, maakt u met INSERT BEFORE een nieuw record 5. Het oude record 5 wordt record 6, enzovoort.

Wanneer u de optie BLANK opgeeft, wordt een nieuw record ingevoegd, maar krijgt u niet de gelegenheid gegevens in dat record in te voeren. Er wordt een leeg record in het database-bestand geplaatst. U kunt de gegevens later invoeren met behulp van het commando BROWSE, CHANGE, EDIT of REPLACE.

Tips

Als u de inhoud van het voorgaande record wilt kopiëren naar het ingevoegde record, moet u SET CARRY op ON instellen voordat u records invoegt. Met SET CARRY TO kunt u tevens opgegeven velden kopiëren.

Speciaal geval

In een netwerk werkt INSERT alleen als het bestand in de modus EXCLUSIVE is geopend.

Voorbeeld

Op de volgende wijze voegt u een nieuw record in voor record 4 (u maakt dus een nieuw record 4) in het database-bestand Afnemers:

```
. USE Afnemers
. GO 4
AFNEMERS: Recordnr.          4
. INSERT BEFORE
```

Zie ook

@, APPEND, BROWSE, CHANGE, EDIT, MODIFY COMMAND, READ, REPLACE, SET CARRY, SET EXCLUSIVE, SET FORMAT

JOIN

Met het commando JOIN maakt u een nieuw database-bestand door bepaalde records en velden uit twee geopende database-bestanden te mengen.

Syntaxis

JOIN WITH <alias> TO <bestandsnaam> FOR <voorwaarde>
[FIELDS <veldenlijst>

Standaardinstelling

Wanneer de bestandsnaam achter TO niet in de actieve bestandsindex staat, moeten een station en een bestandsindex worden opgegeven. De extensie .dbf wordt gebruikt, tenzij u een andere extensie opgeeft.

Gebruik

Wanneer u een actief database-bestand mengt met een geopend bestand uit een ander werkgebied, moet u voor het tweede bestand de alias gebruiken. Het is mogelijk dat de aliasnaam gelijk is aan de bestandsnaam. Als in beide bestanden dezelfde veldnaam voorkomt en u een veld wilt opnemen uit het werkgebied dat niet actief is, plaatst u de aliasnaam voor de veldnaam.

In de veldenlijst mogen alle typen velden uit beide bestanden worden opgegeven, behalve memovelden. Wanneer u probeert memovelden te mengen, verschijnt de foutmelding **Ongeldige bewerking van memoveld**.

In plaats van een veldenlijst op te geven, kunt u ook het commando SET FIELDS geven. Alleen de velden die zijn opgegeven met SET FIELDS, worden in het nieuwe database-bestand opgenomen.

De recordwijzer wordt op het eerste record in het actieve database-bestand geplaatst. Daarna wordt elk record in het tweede bestand getoetst aan de FOR <voorwaarde>. Als de opgegeven voorwaarde waar (.T.) is, wordt een record aan het nieuwe bestand toegevoegd. Nadat alle records in het tweede bestand zijn getoetst, wordt de recordwijzer van het actieve database-bestand op het tweede record geplaatst en wordt het proces herhaald. Dit gaat zo door tot alle records in het actieve bestand zijn verwerkt. Bij grote bestanden kan deze bewerking lang duren.

Wanneer u geen veldenlijst opgeeft, worden eerst velden uit het actieve bestand toegekend. Daarna worden velden uit het tweede bestand toegekend, totdat de grens van 255 velden is bereikt. Veldnamen die in beide te mengen bestanden voorkomen, komen slechts één keer voor in het nieuwe bestand.

Wanneer SET CATALOG op ON is ingesteld, wordt de catalogus automatisch bijgewerkt.

Tips

Gebruik geen letter uit de reeks A tot en met J, of de letter M, als naam voor een database-bestand. Deze letters mogen alleen worden gebruikt als aliasnaam. AA is bijvoorbeeld wel een

geldige naam voor een database-bestand, maar A niet.

Zorg ervoor dat u over voldoende vrije schijfruimte beschikt als u twee bestanden gaat mengen. Wanneer twee database-bestanden worden gemengd, kan het zijn dat er op het station niet genoeg ruimte is voor het nieuwe bestand. Als de voorwaarde niet zorgvuldig wordt opgegeven, kan het nieuwe gemengde bestand behoorlijk groot worden. Wanneer bijvoorbeeld twee bestanden met 1000 records worden gemengd en de opgegeven voorwaarde altijd waar (.T.) is, bevat het nieuwe bestand 1.000.000 records.

Voorbeelden

U kunt de database-bestanden Afnemers en Transact combineren tot een nieuw database-bestand, Nieuwbst, waarin het cliëntnummer, het nummer van de bestelling, de naam van de cliënt, de datum van de bestelling en het bedrag van de bestelling worden opgenomen.

```
. USE Afnemers
. USE Transact IN 2
```

In het volgende voorbeeld is Nieuwbst het nieuwe database-bestand dat wordt gemaakt met de velden Klant_nr, Klantnaam, Dat_trans, Totaal_rek en Bestel_nr uit de database-bestanden Afnemers.dbf en Transact.dbf. Omdat Dat_trans, Totaal_rek en Bestel_nr alleen in Transact staan en Transact zich niet in het actieve werkgebied bevindt, moet u aangeven waar deze velden zich bevinden. U kunt de velden in werkgebied 1 aangeven met de alias A of Afnemers, en de velden in werkgebied 2 met de alias B of Transact.

```
.JOIN WITH Transact TO Nieuwbst FOR Klant_nr=B->Klant_nr FIELDS Klant_nr,  ;
   Klantnaam, B->Dat_trans, B->Totaal_rek, B->Bestel_nr
12 records samengevoegd
. USE Nieuwbst
. DISPLAY STRUCTURE

Structuur van database:  C:\DATA\NEWFILE.DBF
Aantal records      :     12
Laatst gewijzigd op :     23-11-87
    Num   Veldnaam     Veldtype   Breed   Dec    Index
      1   KLANT_NR     Teken        6            J
      2   KLANTNAAM    Teken       30            J
      3   DAT_TRANS    Datum        8            N
      4   TOTAAL_REK   Numeriek     8     2      N
      5   BESTEL_NR    Teken        6            J
** Totaal **                      59
. LIST NEXT 5 Klantnaam, Dat_trans, Bestel_nr

Recordnr.   Klantnaam                Dat_trans   Totaal_rek   Bestel_nr
        1   BALJUW & CO.             09-03-87       415,00    87-109
        2   BALJUW & CO              20-03-87       700,00    87-112
        3   L. G. BLOEM & PARTNERS   10-02-87      1200,00    87-106
        4   L. G. BLOEM & PARTNERS   23-02-87      1250,00    87-108
        5   L. G. BLOEM & PARTNERS   01-04-87       165,00    87-115
```

Zie ook

SET FIELDS, SET RELATION

KEYBOARD

Met het commando KEYBOARD kunt u een tekenreeks in de typebuffer plaatsen. De tekenreeks wordt in dBASE IV vervolgens gelezen als een door de gebruiker getypte tekenreeks.

Syntaxis

KEYBOARD <Tuitdr> [CLEAR]

Gebruik

De tekenreeks in de typebuffer wordt pas gelezen nadat u een commando uitvoert waarbij invoer via het toetsenbord nodig is. Om een aantal toetsen toe te voegen aan commando's zoals BROWSE, EDIT en READ, moet u eerst de toetsen invoeren met het commando KEYBOARD waarna u het commando uitvoert dat de ingevoerde toetsten gebruikt.

KEYBOARD houdt rekening met de standaard capaciteit van de typebuffer, die 20 tekens is. Als de capaciteit wordt overschreden, wordt het teveel aan tekens afgekapt. De capaciteit kan met SET TYPEAHEAD worden gewijzigd.

U kunt alle ASCII-tekens invoeren die door de functie CHR() voor de waarde 1 tot en met 255 worden weergegeven. CHR(0) wordt als invoer genegeerd.

Ook stuurtoetsen zoals ↵, **Home** en **Esc** kunt u in de typebuffer opnemen met de functie CHR(). De waarden voor deze toetsen zijn gelijk aan de waarden die de functie INKEY() oplevert (zie de bespreking van INKEY() in hoofdstuk 4). De waarden die INKEY() voor sommige toetsen oplevert (bijvoorbeeld de functietoetsen en combinaties met de **ALT**-toets), zijn negatief en kunnen daarom niet worden gebruikt bij KEYBOARD() en CHR(). U kunt deze toetsen wel in een macro vastleggen en uitvoeren.

Opties

Met de optie CLEAR worden eventuele ongelezen tekens die nog in de typebuffer staan, gewist voordat er nieuwe tekens worden ingevoerd.

Wanneer u de optie CLEAR niet gebruikt, wordt de buffer niet automatisch gewist.

Speciale gevallen

Speciale ASCII-tekens kunt u invoeren met behulp van de **ALT**-toets en het numerieke toetsenblok.

U kunt bij het commando KEYBOARD een nultekenreeks of een reeks met een lengte 0 invoeren. Het commando wordt dan zonder foutmelding uitgevoerd, maar er wordt niets in de typebuffer ingevoerd.

KEYBOARD

Voorbeeld

Hierna staat een gedeelte van een programma dat een BROWSE-scherm weergeeft met de optie NOCLEAR. Met het commando KEYBOARD wordt **Esc** in de typebuffer ingevoerd. Daardoor wordt BROWSE inactief gemaakt, terwijl de afgebeelde tabel blijft staan.

```
USE Magazijn
KEYBOARD CHR(27)      && Esc vanuit het BROWSE-scherm
BROWSE NOCLEAR
```

Zie ook

CHR(), CLEAR TYPEAHEAD, INKEY(), PLAY MACRO, READKEY(), RESTORE MACROS, SAVE MACROS, SET TYPEAHEAD

LABEL FORM

Met het commando LABEL FORM kunt u op basis van een met MODIFY LABEL ontworpen etiketbestand etiketten afdrukken, weergeven of opslaan.

Syntaxis

LABEL FORM <naam etiketbestand>/?
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]
[SAMPLE] [TO PRINTER/TO FILE <bestandsnaam>]

Standaardinstelling

Wanneer het bestand niet in de actieve bestandsindex staat, moeten de stationsaanduiding en de bestandsindex aan de bestandsnaam voorafgaan. Tenzij u anders bepaalt, krijgt het etiketbestand de extensie .lbl en het gegenereerde etiket de extensie .lbg.

Er worden etiketten afgedrukt voor alle records in het actieve database-bestand, tenzij anders wordt bepaald door het bereik, de clausule FOR of WHILE, of door een actief filter.

Gebruik

Het etiketmodel dat u hebt gemaakt met CREATE/MODIFY LABEL, staat in een bestand met de extensie .lbl. De dBASE-code die op basis daarvan is gegenereerd, staat in een bestand met de extensie .lbg. Met LABEL FORM compileert u een .lbg-bestand tot een .lbo-bestand en voert u dit laatste bestand uit.

Als u het commando LABEL FORM geeft, wordt eerst aan de hand van een aanwijzing in het .lbl-bestand gecontroleerd of het etiketbestand in dBASE III PLUS-indeling of dBASE IV-indeling is opgeslagen.

Als het bestand in dBASE IV-indeling is opgeslagen en er geen .lbo-bestand bestaat, compileert LABEL FORM het .lbo-bestand, waarna de etiketten worden afgedrukt.

Als er wel een .lbo-bestand bestaat en SET DEVELOPMENT op ON is ingesteld, worden datum en tijd van het .lbo- en het .lbg-bestand vergeleken. Als het .lbg-bestand later dan het .lbo-bestand is gemaakt, wordt het .lbg-bestand opnieuw gecompileerd voordat de etiketten worden afgedrukt.

Als er een .lbo-bestand bestaat en SET DEVELOPMENT op OFF is ingesteld, wordt dit bestand uitgevoerd zonder dat datum en tijd worden vergeleken met die van het .lbg-bestand.

Als SET ECHO op ON is ingesteld, wordt broncode uit het .lbg-bestand afgedrukt in plaats van de etiketten.

Opties

Met behulp van de optie SAMPLE kunt u proefetiketten afdrukken, zodat u kunt controleren of de etiketten juist worden uitgelijnd op de printer. Er wordt één rij proefetiketten gemaakt met het teken *x* in plaats van gegevens uit het database-bestand. U kunt dit proces zo vaak als u wilt herhalen door J te antwoorden op de vraag **Wilt u meer voorbeelden?** Wanneer u

antwoordt met N, worden de eigenlijke etiketten afgedrukt.

Met behulp van de optie TO FILE slaat u de etiketten op in een bestand in plaats van de etiketten af te drukken of weer te geven. Als u deze optie hebt opgegeven en een ASCII-printerstuurprogramma hebt geïnstalleerd (Ascii.pr2), wordt er een bestand gemaakt met de extensie .txt. Het .txt-bestand bevat geen escape-codes. Als u een ander printerstuurprogramma hebt geïnstalleerd, wordt er een bestand met de extensie prt gemaakt. In het .prt-bestand kunnen escape-codes voorkomen die specifiek zijn voor de geïnstalleerde printer.

Het vraagteken, ?, wordt de vraagclausule genoemd. Met deze clausule kunt u een menu activeren dat meerdere etiketbestanden bevat.

Speciale gevallen

Bij bewerkingen in een multi-user omgeving wordt, als SET LOCK op ON is ingesteld, het database-bestand door het commando LABEL FORM automatisch vergrendeld voordat het wordt afgedrukt. Als een andere gebruiker het bestand al met RLOCK() of FLOCK() heeft vergrendeld, wordt door LABEL FORM de foutmelding **Bestand in gebruik door een andere gebruiker** gegenereerd. U kunt het bestand naar een tijdelijk bestand kopiëren om het rapport toch te genereren. Het bestand wordt niet vergrendeld als SET LOCK op OFF is ingesteld.

Zie ook

CREATE/MODIFY LABEL, FLOCK(), RLOCK(), SET LOCK, SET PRINTER

In hoofdstuk 5, "Systeemgeheugenvariabelen", worden geheugenvariabelen zoals _pdriver behandeld, die mede bepalen hoe de etiketten worden afgedrukt.

In *Menu's van dBASE IV* wordt de etikettengenerator uitgebreider behandeld.

LIST/DISPLAY

Met het commando LIST/DISPLAY kunt u de inhoud van een database-bestand in kolommen weergeven.

Syntaxis

LIST/DISPLAY [[FIELDS] <uitdrukkingenlijst>] [OFF]
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]
[TO PRINTER/TO FILE <bestandsnaam>]

Standaardinstelling

Met LIST of DISPLAY geeft u alle records weer, tenzij het aantal records wordt beperkt door een bereik, een FOR- of WHILE-clausule, SET FILTER of SET DELETED. Met DISPLAY geeft u alleen het actieve record weer, tenzij er een bereik of een FOR- of WHILE-clausule is opgegeven.

Gebruik

LIST en DISPLAY ALL zijn vrijwel identiek. Het enige verschil is dat DISPLAY na ieder scherm stopt. LIST doet dat niet. Na ieder scherm geeft DISPLAY de melding **Druk op een toets om door te gaan...**

Druk op **Ctrl**-S om LIST of DISPLAY te onderbreken (als SET ESCAPE op ON is ingesteld). U kunt het proces hervatten door op een willekeurige toets te drukken.

Door op **Esc** te drukken kunt u het commando LIST afbreken (als SET ESCAPE op ON is ingesteld).

De inhoud van memovelden wordt niet weergegeven tenzij deze uitdrukkelijk worden genoemd in de veldenlijst. In plaats daarvan verschijnt de veldnaam boven het woord **memo** (kleine letters) als er geen tekst in het memoveld staat, of **MEMO** (hoofdletters) als er wel tekst in het memoveld staat. Wanneer de memovelden worden opgegeven in de veldenlijst, wordt de inhoud weergegeven in kolommen van 50 tekens breed. Met het commando SET MEMOWIDTH kunt u deze standaardinstelling wijzigen.

Wanneer u met LIST of DISPLAY een record weergeeft dat meer dan 80 tekens breed is, loopt de inhoud van het record door op de volgende regel. Een veld kan worden verdeeld over twee of meer regels.

Opties

Met de optie TO PRINTER wordt de uitvoer van het commando afgedrukt, en met de optie TO FILE wordt de uitvoer opgeslagen in een bestand.

Door de optie OFF op te geven laat u de recordnummers weg.

Speciale gevallen

Wanneer SET HEADING op ON is ingesteld, heeft iedere kolom een kopregel. Als de weergegeven waarde de uitkomst is van een uitdrukking waarin een veld voorkomt (bijvoorbeeld Kosten/25), wordt deze uitdrukking de kopregel voor de desbetreffende kolom.

De kopregel wordt net zo weergegeven als u deze typt. Om de kopregel in hoofdletters weer te geven, typt u bijvoorbeeld DISPLAY VELD1. Om de kopregel te laten beginnen met een hoofdletter, typt u DISPLAY Veld1.

Voorbeeld

```
. USE Magazijn
. SET HEADING OFF
. LIST OFF STR(Aantal,3,0) + SPACE(3) + STR(Stukprijs) + " " + Artikel
      FOR Stukprijs >300

  1      1200,00    SOFA,6 POTEN
  1       650,00    SOFA,6 POTEN
  1      1200,00    SOFA,8 POTEN
  1      1250,00    STOEL,BUREAU
  1      1250,00    STOEL,BUREAU
  1      1000,00    STOEL,BUREAU
  2       350,00    STOEL,KANTINE
  1      1500,00    BUREAU,DIRECTIE,5 POTEN
  1      1000,00    STOEL,BUREAU
```

Zie ook

SET ESCAPE, SET HEADING, SET MARGIN, SET MEMOWIDTH

LIST/DISPLAY FILES

Met het commando LIST/DISPLAY FILES vraagt u informatie over de bestandsindex. Deze informatie is vergelijkbaar met de informatie die wordt gegeven door het DOS-commando DIR.

Syntaxis

LIST/DISPLAY FILES [LIKE <structuur>]
[TO PRINTER/TO FILE <bestandsnaam>]

Standaardinstelling

Wanneer u geen bestandsnaam of structuur opgeeft, geeft LIST/DISPLAY FILES alleen informatie over database-bestanden.

Gebruik

In het geval van database-bestanden resulteert dit commando in de bestandsnamen, het aantal records, de datum van de laatste wijziging, de bestandsgrootte (in bytes), het totale aantal weergeven bestanden, het totale aantal bytes van de weergegeven bestanden en het totale aantal bytes dat nog vrij is op het station.

LIST/DISPLAY FILES is een alternatief voor DIR, maar bij DIR zijn de opties TO FILE en TO PRINTER niet beschikbaar. Met LIST/DISPLAY FILES daarentegen kunt u de uitvoer afdrukken of in een bestand opslaan.

Voorbeelden

```
. LIST FILES

Database        Aantal records   Laatste wijziging   Grootte
AFNEMERS.DBF                 8   05-11-87            1570
MAGAZIJN.DBF                17   05-11-87            1569
TRANSACT.DBF                12   05-11-87            554

     3693 bytes in       3 bestanden
8357056 bytes over in station
```

Zie ook

DIR, FILE(), DEFAULT, SET DIRECTORY

LIST/DISPLAY HISTORY

Met het commando LIST/DISPLAY HISTORY vraagt u een lijst van de commando's die zijn uitgevoerd en in de logboekbuffer zijn opgeslagen.

Syntaxis

LIST/DISPLAY HISTORY [LAST <Nuitdr>]
[TO PRINTER/TO FILE <bestandsnaam>]

Standaardinstelling

Standaard kunnen 20 commando's in de logboekbuffer worden opgeslagen. U kunt dit aantal wijzigen met het commando SET HISTORY.

Gebruik

DISPLAY pauzeert na ieder scherm, maar LIST niet.

Met dit commando kunt u zien welke commando's reeds zijn uitgevoerd.

Opties

Alle commando's die in de logboekbuffer staan, worden weergegeven, tenzij u de optie LAST opgeeft. In dat geval worden alleen de laatste <Nuitdr> commando's weergegeven.

Met de optie TO PRINTER wordt de uitvoer van het commando afgedrukt, en met de optie TO FILE wordt de uitvoer opgeslagen in een bestand.

Zie ook

SET HISTORY

LIST/DISPLAY MEMORY

Met het commando LIST/DISPLAY MEMORY vraagt u informatie over de wijze waarop dBASE IV het geheugen van uw computer gebruikt.

Syntaxis

LIST/DISPLAY MEMORY [TO PRINTER/TO FILE <bestandsnaam>]

Gebruik

DISPLAY MEMORY stopt na ieder scherm en geeft de melding **Druk op een toets om door te gaan**... LIST MEMORY stopt niet na ieder scherm.

Als u dit commando geeft, verschijnt er een groot aantal gegevens op het scherm: het totale aantal actieve geheugenvariabelen en elementen, de namen en de status (PUBLIC of PRIVATE) van de verschillende geheugenvariabelen en elementen, de waarden van de geheugenvariabelen en elementen, de namen en definities van alle actieve vensters, pop-up's, menu's en strips en de totale hoeveelheid geheugen die hierdoor wordt gebruikt, de instellingen van alle systeemgeheugenvariabelen en de hoeveelheid geheugen die nog beschikbaar is.

Bij numerieke variabelen wordt zowel de waarde als het type (F of N) weergegeven.

Het aantal beschikbare geheugenvariabelen is het product/de vermenigvuldiging van de instelling van MVBLKSIZE en MVMAXBLKS in het Config.db-bestand. MVBLKSIZE bepaalt het aantal *posities* voor geheugenvariabelen dat een blok mag bevatten. MVMAXBLKS bepaalt het maximale aantal *blokken* dat in het geheugen mag worden toegewezen, waarin geheugenvariabelen kunnen worden geplaatst. Als MVBLKSIZE = 50 en MVMAXBLKS = 10 (de standaardinstelling), kunt u maximaal 10 blokken van 50 posities opslaan, ofwel een totaal van 500 geheugenvariabelen.

De blokken worden toegewezen op het moment dat deze nodig zijn. Wanneer een blok vol is, wordt een nieuw blok toegewezen. Er worden blokken toegewezen totdat het maximale aantal blokken, opgegeven bij MVMAXBLKS, is bereikt. Iedere positie in een blok neemt 56 bytes in beslag. Wanneer het eerste blok wordt toegewezen, reserveert dBASE voldoende ruimte voor 50 geheugenvariabelen, ofwel 2800 bytes.

Iedere reeks die u benoemt, neemt slechts één positie in beslag. Er wordt een speciaal blok toegewezen aan iedere reeks om de reekselementen in op te slaan. De reekselementen worden dus niet opgeslagen in de posities die worden geïnitialiseerd door MVBLKSIZE en MVMAXBLKS.

Als een geheugenvariabele van het type datum, logisch of numeriek is, worden de gegevens in de variabele-positie zelf opgeslagen. Als de geheugenvariabele van het type teken is, wordt er vanuit de positie verwezen naar een ander deel van het geheugen, waarin de tekenreeks zich bevindt.

LIST/DISPLAY MEMORY

Bij elke sessie maakt dBASE runtime-symbolen voor iedere unieke naam van een geheugenvariabele of veld die in een programma of vanaf de commandostip wordt gebruikt. dBASE IV maakt slechts één runtime-symbool voor een veld of geheugenvariabele, ongeacht het aantal keren dat ernaar wordt verwezen. Het maximum aantal runtime-symbolen is het produkt van RTBLKSIZE en RTMAXBLKS in het Config.db-bestand. RTBLKSIZE is het aantal posities voor runtime-symbolen dat in een blok mag staan. RTMAXBLKS is het maximum aantal blokken dat in het geheugen mag worden toegewezen voor runtime-symbolen. De standaardinstelling van RTBLKSIZE is 50. De standaardinstelling van RTMAXBLKS is 10. U kunt dus in totaal 500 runtime-symbolen opslaan.

Net zoals de blokken voor geheugenvariabelen worden de blokken voor runtime-symbolen toegewezen wanneer ze nodig zijn. Als een blok vol is, wordt een nieuw blok toegewezen. De blokken worden toegewezen tot het maximale aantal blokken, dat wordt bepaald door RTMAXBLKS, is bereikt. Elk runtime-symbool neemt 17 bytes in beslag. Als het eerste blok wordt toegewezen, reserveert dBASE voldoende ruimte voor 50 symbolen, ofwel 850 bytes.

Wanneer u een programma uitvoert dat veel verwijzingen naar variabelen of velden bevat, kan het zijn dat u meer geheugen moet toewijzen voor runtime-symbolen. Wanneer er in het Config.db-bestand niet genoeg posities voor runtime-symbolen zijn toegewezen, verschijnt de melding **Maximum aantal symbolen tijdens uitvoering overschreden** wanneer het programma wordt uitgevoerd. U bent dan genoodzaakt de instelling van RTBLKSIZE of RTMAXBLKS te verhogen.

Wanneer geheugenvariabelen worden weergegeven, wordt daarvoor een bepaalde hoeveelheid extra geheugen gebruikt. Datumvariabelen nemen 7 bytes in beslag, logische variabelen 1 byte en numerieke variabelen 20 bytes. Omdat de informatie voor tekenvariabelen zich al in het geheugen bevindt, is er geen extra geheugen nodig om deze variabelen weer te geven. Op de laatste regel van LIST/DISPLAY MEMORY is te zien hoeveel extra geheugen wordt gebruikt om de logische en numerieke variabelen en datumvariabelen weer te geven en om de tekengegevens op te slaan. De systeemgeheugenvariabelen nemen 264 bytes in beslag, dus zelfs als u geen andere variabelen opgeeft, wordt er altijd 264 bytes weergegeven op de laatste regel. **De regel bytes in gebruik voor tekenreeksen geheugenvariabelen** laat zien hoeveel extra geheugen iedere nieuwe variabele in beslag neemt.

Wanneer u extra geheugen nodig hebt voor andere handelingen, zoals het maken van vensters of menu's, kunt u de instelling van MVBLKSIZE, MVMAXBLKS, RTBLKSIZE en RTMAXBLKS in het Config.db-bestand wijzigen.

Opties

Met de optie TO PRINTER wordt de uitvoer van het commando afgedrukt, en met de optie TO FILE wordt de uitvoer opgeslagen in een bestand.

Voorbeelden

De tekenreeks "Hallo" is opgeslagen als de geheugenvariabele Mgroeten, het binair gecodeerde decimale getal 400,34 is opgeslagen als Mvast, de logische waarde waar (.T.) is opgeslagen als Mwaar en het binaire getal 3,21E+06 is opgeslagen als de variabele Mzwevend.

```
. DISPLAY MEMORY
        Geheugenvariabelen gebruiker
MGROETEN      pub   T "Hallo"
MZWEVEND      pub   N            400,34   (400,3400000000000000)
MWAAR         pub   L .T.
MVAST         pub   Z          3210000   (3210000,000000000000)
        4 uit 500 geheugenvariabelen gedefinieerd (en 0 reekselementen)
        .
        .
        .
```

Zie ook

ACCEPT, AVERAGE, CALCULATE, COUNT, DECLARE, DEFINE MENU, DEFINE PAD, DEFINE POPUP, DEFINE WINDOW, INPUT, PRIVATE, PUBLIC, RELEASE, RESTORE, SAVE, STORE, SUM

In hoofdstuk 5, "Systeemgeheugenvariabelen", worden de systeemgeheugenvariabelen behandeld die door dit commando worden weergegeven.

Hoofdstuk 6, "dBASE IV aanpassen", bevat informatie over de instellingen in het bestand Config.db, waaronder MVBLKSIZE, MVMAXBLKS, RTBLKSIZE en RTMAXBLKS.

LIST/DISPLAY STATUS

Met het commando LIST/DISPLAY STATUS vraagt u informatie over de dBASE IV-sessie in uitvoering.

Syntaxis

LIST/DISPLAY STATUS [TO PRINTER/TO FILE <bestandsnaam>]

Gebruik

DISPLAY STATUS stopt na elk scherm en geeft de melding **Druk op een toets om door te gaan**... LIST STATUS stopt niet.

Over elk geopend database-bestand verschijnt de volgende informatie:

- Nummer van het actieve werkgebied
- Naam van database met station, pad en alias
- Status read-only als bestand is voorzien van een schrijfbeveiliging
- Geopende indexbestanden (zowel .ndx- als .mdx-bestanden), de sleuteluitdrukkingen van de indexlabels en aanduiding van de indexen die UNIQUE of DESCENDING zijn
- Geopende memobestanden
- Filterformules
- Database-relaties
- Indelingsbestanden

Verder verschijnen:

- Zoekpad voor bestanden
- Standaardstation
- Actief station en actieve bestandsindex van besturingssysteem
- Afdrukbestemming
- Geladen modules
- Geselecteerd werkgebied
- Instelling van linker marge
- Het momenteel geopende procedurebestand
- REPROCESS-aantal
- REFRESH-interval
- De instelling van DEVICE (SCREEN, PRINT of FILE)
- Valutateken

- Scheidingsteken
- Aantal geopende bestanden
- Instellingen van ON-commando's
- Actieve instelling van de meeste ON/OFF SET-commando's
- Toewijzingen van functietoetsen

Opties

Met de optie TO PRINTER wordt de uitvoer van het commando afgedrukt, en met de optie TO FILE wordt de uitvoer opgeslagen in een bestand.

Speciale gevallen

In een netwerk wordt door LIST/DISPLAY STATUS aangegeven of een geopend database-bestand is vergrendeld. Alle vergrendelde records in de verschillende werkgebieden verschijnen in de lijst.

Zie ook

ALIAS(), DIR, DISKSPACE(), FILE(), INDEX, KEY(), MDX(), MEMORY(), NDX(), ORDER(), OS(), PROGRAM(), SET, TAG(), VERSION()

LIST/DISPLAY STRUCTURE

Met het commando LIST/DISPLAY STRUCTURE vraagt u om de velddefinities van het opgegeven database-bestand.

Syntaxis

LIST/DISPLAY STRUCTURE [IN <alias>]
[TO PRINTER/TO FILE <bestandsnaam>]

Gebruik

Als u dit commando geeft, verschijnen de volgende gegevens: de bestandsnaam, het aantal records, de datum van de laatste wijziging, de volledige definities van de velden, de velden die indexlabels zijn in het bijbehorende .mdx-bestand en het totale aantal bytes in een record.

Het totale aantal bytes in een record is de som van alle veldlengten plus één (de extra byte is gereserveerd voor een eventuele wismarkering). De bestandsaanhef draagt ook bij tot het totale aantal bytes in een record.

Wanneer een database meer velden heeft dan op het scherm passen (waarbij het actieve scherm een kleine venster kan zijn of het volledige scherm met 43 regels), stopt DISPLAY STRUCTURE steeds nadat het scherm met velden is gevuld en verschijnt de melding **Druk op een toets om door te gaan**... LIST STRUCTURE stopt niet.

Wanneer SET FIELDS op ON is ingesteld, verschijnt het symbool > naast ieder veld dat is opgegeven in het commando SET FIELDS TO.

Opties

Wanneer u IN <alias> opgeeft, wordt de structuur van het opgegeven werkgebied weergegeven. Met de clausule IN kunt u de structuur van een ander werkgebied bekijken zonder het werkgebied te kiezen met het commando SELECT.

Met de optie TO PRINTER wordt de uitvoer van het commando afgedrukt, en met de optie TO FILE wordt de uitvoer opgeslagen in een bestand.

Voorbeeld

Op de volgende wijze geeft u de structuur van de database Afnemers weer:

```
. USE Afnemers
. LIST STRUCTURE

Structuur van database:     C:\DBASE\AFNEMERS.DBF
Aantal records:     8
Laatst gewijzigd op:     05-11-87
Num     Veldnaam        Type    Breedte    Dec    Index
  1     KLANT_NR        Teken         6            J
  2     KLANTNAAM       Teken        30            J
  3     ACHTERNAAM      Teken        15            N
  4     VOORNAAM        Teken        15            N
  5     ADRES           Teken        30            N
  6     WOONPLAATS      Teken        20            N
  7     PROVINCIE       Teken         2            N
  8     POSTCODE        Teken        10            N
  9     TELEFOON        Teken        13            N
 10     OVERZICHT       Memo         10            N
** Totaal **                        152
```

Zie ook

COPY STRUCTURE, COPY STRUCTURE EXTENDED, CREATE FROM, CREATE/MODIFY STRUCTURE, SET FIELDS

LIST/DISPLAY USERS

Met het commando LIST/DISPLAY USERS vraagt u op, welke werkstations in het netwerk zijn aangemeld bij dBASE IV.

Syntaxis

LIST/DISPLAY USERS

Gebruik

Met het commando LIST/DISPLAY USERS wordt het bestand Login.db in de bestandsindex DBNETCTL.300 gelezen, en worden hieruit de namen gehaald, waaronder de werkstations bij het netwerk zijn aangemeld.

Speciale gevallen

Geef voordat u dBASE IV de-installeert, altijd het commando LIST/DISPLAY USERS, of een soortgelijk commando van het netwerkbesturingssysteem, om te controleren of iemand dBASE IV nog gebruikt.

Wanneer twee of meer gebruikers zich aanmelden met dezelfde werkstationsnaam, wordt de gebruikersnaam maar één keer weergegeven door het commando LIST/DISPLAY USERS. Wanneer bijvoorbeeld twee gebruikers zich aanmelden als WKSTN1, wordt dit als volgt weergegeven:

Computer-naam

>WKSTN1

zelfs wanneer de gebruikers zich in dezelfde bestandsindex hebben aangemeld.

Zie ook

LIST/DISPLAY STATUS, NETWORK()

LOAD

Met LOAD kunt u binaire programmabestanden in het geheugen laden.

Syntaxis

LOAD <binaire bestandsnaam>

Gebruik

Met LOAD wordt een binair (.bin) bestand in het geheugen geplaatst, waar het kan worden uitgevoerd met het commando CALL of de functie CALL().

Opties

De <binaire bestandsnaam> kan een bestandsnaam zijn of een tekenuitdrukking die een bestandsnaam oplevert.

Richtlijnen voor het programmeren

In dBASE IV wordt elk geladen bestand beschouwd als subroutine of programmamodule en niet als een extern programmabestand.

Maximaal kunnen 16 bestanden tegelijkertijd in het geheugen worden geladen. Elk bestand mag 32.000 bytes groot zijn en moet een binair bestand zijn.

Elke geladen module moet een unieke naam hebben. De standaardextensie is .bin. Wanneer u een bestand aanroept met CALL, hoeft u de extensie niet op te geven. De bestandsnaam wordt de naam van de module.

Als u een bestand laadt dat dezelfde naam heeft als een eerder geladen bestand maar een andere extensie, wordt het oude bestand in het geheugen vervangen door het nieuwe.

Met het commando LIST/DISPLAY STATUS kunt u de namen van geladen modules weergeven.

De betrouwbaarheid van de bestanden die u laadt, wordt door dBASE IV niet gecontroleerd. Controleer zelf of de programma's binair zijn en geen fouten bevatten.

Binaire bestanden die met dBASE III PLUS zijn gemaakt, kunnen in dBASE IV worden gebruikt als in die bestanden gebruik wordt gemaakt van één tekenreeksparameter die eindigt op een nulteken. Binaire bestanden die afhankelijk zijn van de manier waarop in dBASE III PLUS geheugenvariabelen in het geheugen worden geordend, kunnen niet in dBASE IV worden gebruikt.

Bij het ontwerpen van het assembleertaalprogramma voor het binaire bestand moet aan de volgende vereisten worden voldaan:

- De eerste uit te voeren instructie moet met het commando ORG worden gegeven bij offset nul.
- Het programma mag geen geheugen toewijzen of gebruiken dat buiten de opgegeven grenzen valt, omdat LOAD aan de hand van de grootte van het bestand bepaalt hoeveel geheugen moet worden toegewezen.

- Geheugenvariabelen die als argumenten bij CALL of CALL() worden gebruikt, mogen niet door het programma worden verkleind of vergroot.
- Alvorens terug te keren naar dBASE IV, moeten de registers Code Segment (CS) en Stack Segment (SS) worden hersteld.
- Het programma moet naar dBASE IV terugkeren met een far return (RET FAR). De meeste commerciële programma's eindigen met een exit-instructie in plaats van een far return. Voer deze programma's uit met het commando RUN/! en niet met LOAD en CALL.

Met het commando CALL of de functie CALL() kunt u maximaal zeven parameters aan het binaire bestand doorgegeven. De parameters kunnen veldnamen, uitdrukkingen, geheugenvariabelen of reekselementen van elk willekeurig gegevenstype zijn, behalve van het type memo.

Voordat de besturing door dBASE IV aan het binaire programma wordt overgedragen, worden alle parameters berekend en wordt de uitkomst naar een tekenreeks geconverteerd die eindigt op een nulteken.

Het register CX bevat het aantal parameters dat is doorgegeven. DS:BX bevat het adres van de eerste parameter. ES:DI wijst naar een geheugenblok van 28 bytes dat bestaat uit een lijst met zeven adressen van elk vier bytes. De eerste vier bytes wijzen naar de eerste parameter, de volgende vier bytes naar de tweede parameter enzovoort. Als er minder dan zeven parameters zijn overgedragen, wijzen de resterende adressen naar lege tekenreeksen die eindigen op nultekens.

In het binaire programma kan de inhoud van alle parameters worden gewijzigd. Alleen wijzigingen aan parameters die als geheugenvariabelen zijn doorgegeven, blijven behouden wanneer de besturing weer wordt teruggegeven. De geheugenvariabelen die door tekenreeksen werden voorgesteld, worden door dBASE IV weer geconverteerd naar het oorspronkelijke gegevenstype, waarbij de oorspronkelijke gegevenslengte blijft behouden.

Om een assembleertaalprogramma dat is geschreven in een 8086/8088-assembleertaal gereed te maken voor gebruik in dBASE, moet u een assembleerprogramma gebruiken om het bronbestand (.asm) tot een objectbestand (.dbo) te assembleren, het objectbestand aan een uitvoerbaar bestand (.exe) te koppelen en vervolgens van het uitvoerbare bestand een binair bestand te maken. Voor de Macro Assembler van Microsoft (MASM) gebruikt u de volgende commando's:

```
MASM <bron>;

LINK <doel>;

EXE2BIN <doel>
```

Voorbeelden

In het volgende voorbeeld ziet u een assembleerprogramma dat een opgegeven teken op alle plaatsen waar het voorkomt vervangt door een ander opgegeven teken. Het programma staat (evenals het bestand Strsubst.bin) op uw diskette met voorbeeldbestanden.

```
; Strsubst.asm (Bronbestand van Strsubst.bin)
;
; Vervang de tekens in een tekenreeks.
;
CODE     SEGMENT  BYTE   PUBLIC 'CODE'
STRSUB   PROC     FAR
ASSUME   CS:CODE
         PARAM1   EQU    ES:  [DI+0]
         PARAM2   EQU    ES:  [DI+4]
         PARAM3   EQU    ES:  [DI+8]
         PARAM4   EQU    ES:  [DI+12]
         PARAM5   EQU    ES:  [DI+16]
         PARAM6   EQU    ES:  [DI+20]
         PARAM7   EQU    ES:  [DI+24]
START:
   PUSH      BP                    ; Sla het stack-frame op
   MOV       BP, SP
; Stop als er niet ten minste 3 parameters zijn.
   CMP       CX, 3
   JL        DONE
; Laad de eerste byte van de tweede parameter in CL
   LDS       BX, PARAM2            ; DS:BX duidt op de tweede parameter
   MOV       CL, [BX]              ; Sla de eerste byte op
; Laad de eerste byte van de derde parameter in CH
   LDS       BX, PARAM3            ; DS:BX duidt op de derde parameter
   MOV       CH, [BX]              ; Sla de eerste byte op in CH
; Wijs met DS:BX de eerste parameter aan
   LDS       BX, PARAM1
; Maak voor elk teken in de eerste parameter een lus
AGAIN:   MOV     AL, [BX]          ; Breng het volgende teken in AL
         CMP     AL, 0             ; Is dit het einde van de reeks?
         JE      DONE              ;   Ja: stoppen
         CMP     AL, CL            ; Is dit het teken waarnaar wordt gezocht?
         JNE     NEXT              ;   Nee:  niet vervangen
         MOV     [BX], CH          ;   Ja: vervangen
NEXT:    INC     BX                ; Wijs het volgende teken aan
         JMP     AGAIN             ;
DONE:    POP     BP                ; Herstel het stack-frame
   RET
STRSUB   ENDP
CODE     ENDS
         END
```

In dit voorbeeld wordt het binaire bestand Strsubst.bin uitgevoerd.

```
. LOAD Strsubst
. Mstr1= "* * * * * *"
. CALL Strsubst WITH Mstr1, "*", "#"
. ? Mstr1
######
```

Zie ook

CALL, CALL(), LIST/DISPLAY STATUS, RELEASE, RUN, RUN()

LOCATE

Met LOCATE wordt in het actieve database-bestand gezocht naar een record dat voldoet aan de opgegeven voorwaarde.

Syntaxis

LOCATE [FOR] <voorwaarde> [<bereik>] [WHILE <voorwaarde>]

Gebruik

Met het commando LOCATE wordt het database-bestand sequentieel doorzocht en wordt bij ieder record gecontroleerd of het voldoet aan de voorwaarde FOR. Voor ieder record resulteert de voorwaarde in waar (.T.) of onwaar (.F). Als de voorwaarde resulteert in waar, wordt de recordwijzer op het eerste record dat aan de voorwaarde voldoet, geplaatst.

LOCATE werkt ook met database-bestanden die niet zijn geïndexeerd. FIND en SEEK werken alleen met geïndexeerde database-bestanden en zijn over het algemeen sneller. Als LOCATE wordt gebruikt voor een geïndexeerd bestand, wordt de indexvolgorde gebruikt. Als een index wordt gebruikt, kunt u beter het commando SEEK dan het commando LOCATE gebruiken omdat met SEEK in de volgorde van de index wordt gezocht. Wanneer u een database wilt doorzoeken op basis van een ander veld dan het sleutelveld, sluit u eerst alle indexen. Omdat LOCATE de gegevens ongewijzigd laat, kunt u SET INDEX instellen op OFF, LOCATE gebruiken en SET INDEX weer instellen op ON zonder dat de index hoeft te worden bijgewerkt.

Geef een logische voorwaarde op in de FOR-clausule, bijvoorbeeld LOCATE FOR Achternaam = "Gorsman" of LOCATE FOR .T.

Met het commando CONTINUE kunt u het volgende record zoeken dat aan de voorwaarde voldoet. Zelfs als u meerdere LOCATE-commando's geeft voor hetzelfde database-bestand, gaat CONTINUE verder met het laatst gegeven LOCATE-commando.

LOCATE en CONTINUE gelden alleen in het werkgebied waarin deze commando's zijn gegeven. Voor ieder werkgebied kunt u een ander LOCATE-commando geven. Als u een LOCATE-commando geeft, een ander werkgebied kiest, later terugkeert naar het eerste werkgebied en een CONTINUE-commando geeft, wordt verder gezocht waar het eerder gegeven LOCATE-commando in het betreffende werkgebied is opgehouden.

Recordwijzer

Tenzij anders wordt bepaald door het bereik of door een WHILE-clausule, wordt de recordwijzer met LOCATE aan het begin van het database-bestand geplaatst en wordt het zoekcommando vanaf het eerste record uitgevoerd.

Met de optie NEXT <n> kunt u een bepaald aantal records doorzoeken. Met de opties NEXT <n> en REST wordt de recordwijzer niet aan het begin van het database-bestand geplaatst. Het zoekcommando begint in het actieve record.

Als een record aan de voorwaarde voldoet, wordt de recordwijzer op het betreffende record geplaatst. Als SET TALK op ON is ingesteld, wordt het recordnummer weergegeven. FOUND() resulteert in waar (.T.).

Als geen record wordt gevonden dat aan de voorwaarde voldoet, wordt de recordwijzer aan het einde van het bestand geplaatst (EOF() is .T.), of aan het einde van het bereik, als dat is opgegeven. De melding **Einde van LOCATE-bereik** wordt weergegeven. FOUND() resulteert in onwaar (.F.).

Voorbeelden

Om het eerste record in het database-bestand Transact te vinden waarvan het veld Klant_nr de waarde L00001 heeft, geeft u het volgende commando:

```
. USE Transact
. LOCATE FOR Klant_nr = "L00001"
Record =                5
```

Om elk record met het klantnummer C00001 te vinden dat nog niet is gefactureerd, geeft u het volgende commando:

```
. LOCATE FOR Klant_nr = "C00001" .AND. .NOT. Factuur
Record =      11
. DISPLAY
Recordnr.     KLANT_NR      BESTEL_NR     DAT_TRANS     FACTUUR       TOTAAL_REK
     11       C00001        87-115        01-04-87      .F.           165
. CONTINUE
Einde van LOCATE-bereik
. ? EOF()
.T.
. ? FOUND()
.F.
```

Zie ook

CONTINUE, FIND, FOUND(), SEEK, SEEK()

LOGOUT

Met LOGOUT wordt de gebruiker afgemeld en wordt een nieuw aanmeldingsscherm weergegeven.

Syntaxis

LOGOUT

Gebruik

Met het commando LOGOUT kunt u de aan- en afmeldingsprocedures van gebruikers beheren. Met LOGOUT moet de gebruiker zich afmelden en wordt om een aanmelding gevraagd.

Wanneer het commando is uitgevoerd, wordt het scherm van het werkstation leeggemaakt en verschijnt een aanmeldingsscherm. De gebruiker kan een groepnaam, aanmeldingsnaam en wachtwoord invoeren. Met het commando PROTECT worden controlefuncties ingesteld waarmee de aanmelding wordt gecontroleerd en waarmee wordt bepaald tot welk niveau de gebruiker toegang heeft.

Met LOGOUT worden alle geopende database-bestanden, bijbehorende bestanden en programmabestanden gesloten.

Speciale gevallen

Als het PROTECT-commando niet is gegeven, wordt het bestand Dbsystem.db niet gemaakt. De gebruiker keert terug naar de commandostip in plaats van naar het aanmeldingsscherm.

Zie ook

PROTECT, QUIT

MODIFY COMMAND/FILE

Met MODIFY COMMAND/FILE wordt de schermgrote tekstverwerker van dBASE IV of een opgegeven tekstverwerker aangeroepen.

Syntaxis

MODIFY COMMAND/FILE <bestandsnaam> [WINDOW <vensternaam>]

Standaardinstellingen

Als u met MODIFY COMMAND geen station, bestandsindex of extensie opgeeft als deel van de bestandsnaam, worden het standaardstation, de standaardbestandsindex en de extensie .prg gebruikt.

In MODIFY FILE, de alternatieve syntaxis voor de tekstverwerker van dBASE IV, wordt geen standaardextensie gebruikt. Als u met MODIFY FILE een bestand maakt en geen extensie opgeeft, wordt ook geen extensie gebruikt.

De standaardregellengte is 1024 tekens of 1024 kolommen.

Gebruik

De standaardinstelling is dat met MODIFY COMMAND/FILE de schermgrote tekstverwerker van dBASE IV wordt aangeroepen om programmabestanden, indelingsbestanden of standaard ASCII-tekstbestanden te maken en te bewerken. U kunt het programma aanroepen vanaf de commandostip of vanuit het Control Center om een programma- of procedurebestand of een memoveld te bewerken.

Met MODIFY COMMAND wordt het actieve venster gebruikt. U kunt het commando met de optie WINDOW ook aan een ander venster toewijzen. Als u een venster opgeeft dat nog niet met het commando ACTIVATE WINDOW is geactiveerd of dat met het commando CLEAR WINDOWS is gesloten, ziet u de foutmelding **Vensternaam niet gevonden**.

In MODIFY COMMAND/FILE wordt de maximumlengte van bestanden meestal bepaald door de beschikbare geheugenruimte. De maximumomvang van bestanden is 32.000 regels van 1024 tekens, dat wil zeggen dat de maximale bestandsomvang 32.768.000 bytes bedraagt.

In plaats van de tekstverwerker van dBASE IV kunt u ook een andere tekstverwerker gebruiken. U geeft dit tijdens de installatie op in het bestand Config.db. Raadpleeg voor meer informatie de installatie-instructies in hoofdstuk 6.

Wanneer u het commando MODIFY COMMAND/FILE geeft, wordt het opgegeven bestand gezocht. Als het bestaat, wordt het aangeroepen om te worden bewerkt. Als het bestand niet wordt gevonden, wordt een nieuw bestand gemaakt. Met MODIFY FILE kunt u een ASCII-bestand bewerken door de extensie op te geven. Telkens wanneer u een bestand bewerkt, wordt van de oude versie een reservebestand opgeslagen in dezelfde bestandsindex als het oorspronkelijke bestand. Onafhankelijk van de instelling van SET SAFETY wordt een oude versie van een reservebestand steeds automatisch overschreven wanneer het oorspronkelijke bestand wordt gewijzigd.

MODIFY COMMAND/FILE

De naam van een reservebestand is dezelfde als die van het oorsronkelijke bestand. De extensie van het reservebestand is afhankelijk van de extensie van het oorspronkelijke bestand en varieert zoals in de volgende tabel is beschreven:

Tabel 2-7 Extensies van reservebestanden

Extensie originele bestand	Extensie reservebestand
.prg	.bak
.txt	.bak
.dbf	.dbk
.dbt	.tbk
.mdx	.mbk

WAARSCHUWING
Gebruik voor uw bestanden niet de extensie van een reservebestand. Na het bewerken van een bestand met een extensie van een reservebestand wordt er geen reservekopie gemaakt. Ook wordt u niet gewaarschuwd wanneer uw originele bestand wordt overschreven.

Nadat u een .prg-bestand hebt bewerkt, verwijdert de tekstverwerker het oude .dbo-bestand. U moet een nieuw .prg-bestand compileren om een nieuw .dbo-bestand te kunnen maken.

In het scherm MODIFY COMMAND wordt een balkmenu weergegeven met daarin de commando's die u bij het bewerken kunt gebruiken. Een complete lijst met toetsen die u voor het bewerken kunt gebruiken, vindt u in de *Naslaggids*. Meer informatie over het gebruik van de menu's **Opmaak**, **Tekst** en **Afdrukken** vindt u in *Menu's van dBASE IV*.

Een bestand dat is gemaakt met MODIFY COMMAND/FILE, kunt u met het volgende commando afdrukken:

```
. TYPE <bestandsnaam> TO PRINT
```

Zie ook

COMPILE, CREATE, DO, NOTE, RENAME, SET DEVELOPMENT, TYPE

MODIFY COMMAND/FILE

De volgende commando's zijn synoniem met de overeenkomstige CREATE-commando's. Zie het commando CREATE voor de juiste syntaxis en het gebruik van de CREATE-varianten.

MODIFY APPLICATION
MODIFY LABEL
MODIFY QUERY/VIEW
MODIFY REPORT
MODIFY SCREEN
MODIFY STRUCTURE

MOVE WINDOW

Met het commando MOVE WINDOW wordt een venster naar een nieuwe lokatie op het scherm verplaatst.

Syntaxis

MOVE WINDOW <vensternaam> TO <rij>,<kolom>/BY
<delta-rij>,<delta-kolom>

Gebruik

In de syntaxis zijn twee manieren gebruikt om een venster te verplaatsen. U kunt de nieuwe coördinaten voor het venster geven, of u kunt opgeven hoeveel u het venster wilt verplaatsen ten opzichte van de actieve positie.

Wanneer u een venster hebt verplaatst, worden de coördinaten van de nieuwe lokatie toegewezen aan de naam van het betreffende venster. Als u het venster op de oorspronkelijke positie wilt terugplaatsen, geeft u het commando MOVE WINDOW en de oorspronkelijke coördinaten opnieuw.

Als een venster op de nieuwe coördinaten niet op het scherm past, ziet u een foutmelding en wordt het commando MOVE WINDOW niet uitgevoerd.

Voorbeelden

Om het venster W1 twee kolommen naar rechts en vijf regels naar beneden te verplaatsen, geeft u het volgende commando:

```
. MOVE WINDOW W1 BY 5,2
```

Om het venster W1 naar nieuwe coördinaten te verplaatsen, geeft u het volgende commando:

```
. MOVE WINDOW W1 TO 10,5
```

Zie ook

ACTIVATE WINDOW, DEACTIVATE WINDOW, DEFINE WINDOW, RESTORE WINDOW, SAVE WINDOW, SET WINDOW OF MEMO

NOTE

Met NOTE, een asterisk (*) of een dubbele ampersand (&&) worden commentaarregels in een programmabestand (.prg) aangeduid.

Syntaxis

NOTE/* <tekst>

en

[<commando>] &&<tekst>

Gebruik

Met het commando NOTE/*/&& worden commentaarregels in een programma geplaatst die dienen als uitleg en documentatie. U kunt overal in een programma de drie verschillende methoden gebruiken om commentaar aan te duiden. Als een NOTE- of &&-regel eindigt met een puntkomma, wordt de volgende regel als deel van de commentaarregel beschouwd.

Voorbeelden

In dit voorbeeld kunt u zien hoe u NOTE kunt gebruiken:

```
NOTE Dit is een eenvoudige lus
STORE 1 to Tel
DO WHILE Tel < 100
    STORE Tel + 1 TO Tel
ENDDO
```

In dit voorbeeld wordt && gebruikt om commentaar in een programmabestand op te nemen:

```
Mgehvar = 12        && initialiseert numerieke geheugenvariabele
```

Zie ook

MODIFY COMMAND/FILE, PROCEDURE

ON ERROR/ESCAPE/KEY

Met ON ERROR/ESCAPE/KEY wordt een programma uitgevoerd zodra zich een fout voordoet, op Esc wordt gedrukt of op een opgegeven toets wordt gedrukt.

Syntaxis

ON ERROR <commando>
    /ESCAPE <commando>
    /KEY [LABEL<toetsnaam>] [<commando>]

Gebruik

Met het commando ON kunt u een bepaald commando uitvoeren zodra aan een opgegeven voorwaarde wordt voldaan. Deze voorwaarde kan een fout in dBASE IV zijn, het gebruik van **Esc**, het gebruik van een toets die is opgegeven met <toetsnaam> of het gebruik van een willekeurige toets als geen toetsnaam is opgegeven. De fouten in dBASE IV zijn onder meer syntaxisfouten en evaluatiefouten. De voorwaarde ON blijft van kracht totdat u deze verwijdert door ON ERROR/ESCAPE/KEY zonder commando op te geven, of totdat u dBASE IV verlaat. Denk eraan dat ON ESCAPE niet werkt als SET ESCAPE op OFF is ingesteld. ON KEY werkt niet wanneer SET TYPEAHEAD op 0 is ingesteld.

Als ON ERROR wordt geactiveerd en een ON ERROR-commando of -procedure wordt uitgevoerd, wordt ON ERROR inactief gemaakt totdat het commando of de procedure gereed is. U kunt echter binnen de procedure opnieuw een ON ERROR-commando geven. U kunt dus meerdere ON ERROR-commando's nesten die elk inactief worden gemaakt als het commando of de procedure is uitgevoerd.

Met de volgende schermgrote commando's kunnen fouten intern worden onderschept en hersteld. Daarom wordt het <commando> dat bij ON ERROR hoort, niet uitgevoerd als in een van deze commando's een fout optreedt.

APPEND
ASSIST
BROWSE
EDIT
CREATE/MODIFY APPLICATION/LABEL/REPORT/SCREEN/
    STRUCTURE/VIEW
MODIFY COMMAND/FILE

Richtlijnen voor het programmeren

dBASE IV reageert in de volgende volgorde op de ON-opties:

ON ERROR = in dBASE IV heeft zich een fout voorgedaan
ON ESCAPE = u hebt op **Esc** gedrukt
ON KEY = u hebt op de opgegeven toets gedrukt (of, als geen toetsnaam is opgegeven, op een willekeurige toets)

Als bijvoorbeeld ON KEY (zonder toetsnaam) en ON ESCAPE tegelijkertijd actief zijn, wordt het commando ON ESCAPE uitgevoerd als op **Esc** wordt gedrukt. ON KEY wordt niet uitgevoerd.

Als u **Esc** wilt gebruiken in combinatie met het commando ON KEY, moet SET ESCAPE op OFF zijn ingesteld.

Als het commando ON KEY (zonder toetsnaam) actief is en als u op een andere toets dan **Esc** drukt, wordt ON KEY uitgevoerd zodra het actieve commando is uitgevoerd. Als u bijvoorbeeld op een toets drukt terwijl een bestand wordt geïndexeerd, wordt het INDEX-commando niet onderbroken. Als de opgeslagen toets echter niet door het programma wordt verwijderd, wordt het bij ON KEY opgegeven commando herhaalde malen uitgevoerd.

ON ERROR reageert niet op fouten in het besturingssysteem, zoals een niet beschikbare printer of een station dat niet toegankelijk is. Het commando reageert alleen op fouten in dBASE IV, bijvoorbeeld een syntaxisfout of een evaluatiefout. Bovendien worden na het beëindigen van een ON ERROR-commando IF- of DO WHILE-voorwaarden niet opnieuw berekend in het programma waarin de fout zich heeft voorgedaan (tenzij RETRY wordt gebruikt). Als u in een foutprocedure wilt verdergaan met de volgende regel, gebruikt u hiervoor het commando RETURN. Als de volgende regel een DO WHILE/IF-lus is, wordt met RETURN de lus voortgezet.

Met de commando's ON ERROR en ON ESCAPE kunnen weer andere commando's worden uitgevoerd, tenzij deze al zijn geactiveerd binnen een door de gebruiker gedefinieerde functie of het commando ON KEY, ON READERROR of ON PAGE.

U dient te vermijden dat een dBASE-commando herhalend wordt gebruikt bij het commando ON KEY. Raadpleeg het gedeelte over het commando FUNCTION voor een lijst met deze commando's. De beperkingen voor herhalende commando's zijn van toepassing op door de gebruiker gedefinieerde functies, op ON KEY en op ON ESCAPE.

Met de optie LABEL kunt u in het ON KEY-commando specifieke toetsen opgeven in plaats van alle toetsen, zoals in dBASE III PLUS gebruikelijk is. Als u een tekengeheugenvariabele opgeeft bij <toetsnaam>, moet u het macro-substitutiesymbool (&) gebruiken. Als u geen toetsnaam opgeeft, wordt het commando ON KEY LABEL voor elke toets op het toetsenbord gegeven. Als u alleen ON KEY opgeeft zonder de optie LABEL, schakelt u de functie uit.

Wanneer bij SET FUNCTION en ON KEY dezelfde toets is opgegeven, heeft ON KEY een hogere prioriteit dan ON FUNCTION. Als ON KEY op een toets reageert terwijl u gegevens voor een GET-variabele invoert, wordt het commando voor de betreffende toets uitgevoerd. De toets waarmee ON KEY werd geactiveerd wordt niet opgeslagen in de GET-variabele.

ON ERROR/ESCAPE/KEY

Als een WAIT-commando actief is, heeft dit een hogere prioriteit dan ON KEY. De geheugenvariabele die in WAIT wordt gebruikt, ontvangt de invoer onafhankelijk van de bij ON KEY opgegeven toetsen.

ON KEY wordt direct uitgevoerd in EDIT, BROWSE, READ, door de gebruiker gedefinieerde menu's en pop-up menu's of aan het einde van een commando. U kunt stuur- en niet-stuurtoetsen opgeven. Bepaalde commando's met een specifieke taak, zoals LIST, SORT en INDEX, worden niet onderbroken door ON KEY. ON KEY-commando's worden pas uitgevoerd nadat deze commando's zijn uitgevoerd.

Het is niet van belang of de <toetsnaam> bij ON KEY in hoofdletters of kleine letters wordt opgegeven. Dat geldt voor afdrukbare en niet-afdrukbare tekens.

Voor afdrukbare toetsen is de <toetsnaam> het teken, cijfer of symbool dat u ziet. In tabel 2-8 ziet u de toetsnamen van de niet-afdrukbare toetsen op het toetsenbord.

Het commando READ heeft prioriteit op ON ESCAPE. Als READ actief is en u op **Esc** drukt, wordt de volgende regel code na READ uitgevoerd. De toets **Esc** (27) die volgt op READ, kunt u onderscheppen met de functie LASTKEY() of READKEY().

Tabel 2-8 <toetsnaam>-omschrijvingen voor niet-afdrukbare toetsen

Toetsenomschrijving	**<toetsnaam>**
F1 tot en met **F10**	F1, F2, F3 . . .
Ctrl-F1 tot en met **Ctrl-F10**	Ctrl-F1, Ctrl-F2 . . .
Shift-F1 tot en met **Shift-F9**	Shift-F1, Shift-F2 . . .
Alt-0 tot en met **Alt-9**	Alt-0, Alt-1, Alt-2 . . .
Alt-A tot en met **Alt-Z**	Alt-A, Alt-B, Alt-C . . .
←	LEFTARROW
→	RIGHTARROW
↑	UPARROW
↓	DNARROW
Home	Home
End	End
PgUp	PgUp
PgDn	PgDn
Del	Del
Spatie-terug	BACKSPACE

Ctrl-←	Ctrl-leftarrow
Ctrl-→	Ctrl-rightarrow
Ctrl-Home	Ctrl-Home
Ctrl-End	Ctrl-End
Ctrl-PgUp	Ctrl-PgUp
Ctrl-PgDn	Ctrl-PgDn
Ins	INS
Tab	TAB
Shift-Tab	BACKTAB
* **Ctrl-A** tot en met **Ctrl-Z**	Ctrl-A, Ctrl-B, Ctrl-C . . .

* Op de meeste toetsenborden kan de toets ↵ worden onderschept met ON KEY LABEL **Ctrl-M**.

Voorbeelden

In de volgende voorbeelden wordt ervan uitgegaan dat een procedurebestand met Bij_fout en Stop al is geopend.

Geef de volgende opdracht om de optie ON ERROR te activeren als zich een fout voordoet in dBASE IV:

```
. ON ERROR DO Bij_fout
```

Als een fout wordt gevonden, wordt de instructie vervolgd:

```
PROCEDURE Bij_fout
* commando's foutenverwerking
.
.
RETURN
```

Voer het volgende in om met de optie ON KEY het commando DO <programma> uit te voeren, waarmee het afdrukken wordt gestopt en wordt vervolgd met een procedure met de naam Stop:

```
. ON KEY DO Stop
```

ON ERROR/ESCAPE/KEY

Als u op een willekeurige toets drukt, wordt het programma vervolgd met de Stop-procedure. In de volgende procedure wordt u gevraagd de letter A te typen om het afdrukken te stoppen. Als u op een andere toets drukt, wordt het afdrukken vervolgd.

```
PROCEDURE Stop
* Verwijder eerst de toets voor het activeren van ON KEY()
* uit de typebuffer met INKEY().
i = INKEY()
WAIT   "Druk op S om afdrukken te stoppen, " +;
       "of op een nadere toets om door te gaan" TO choice
IF UPPER(choice) = "S"
    RETURN TO MASTER
ENDIF
RETURN
```

Zie ook

INKEY(), LASTKEY(), ON READERROR, PROCEDURE, READKEY(), RETRY, RETURN, SET ESCAPE, SET PROCEDURE, SET TYPEAHEAD, UPPER(), WAIT

ON PAD

Met het commando ON PAD wordt een pop-up menu gecombineerd met een strip (pad) van een opgegeven balkmenu.

Syntaxis

ON PAD <stripnaam> OF <menunaam> [ACTIVATE POPUP <pop-up naam>]

Gebruik

Het pop-up menu dat in het commando ON PAD is opgegeven, wordt geactiveerd als de cursor op de strip van het opgegeven balkmenu staat.

Als u de optie ACTIVATE POPUP niet gebruikt, wordt het keuzevenster uitgeschakeld.

Met de optie ACTIVATE POPUP kunt u balkmenu's en pop-up menu's tegelijkertijd op het scherm activeren. Terwijl u → en ← gebruikt om de cursor in het balkmenu te verplaatsen, worden verschillende pop-up menu's geactiveerd. U kunt de cursor met ↑ en ↓ tussen de balken in de pop-up menu's verplaatsen.

U kunt dit commando niet geven als u het commando ON SELECTION PAD al hebt gegeven om een aanwijzing of een programma in combinatie met een strip te gebruiken.

Voorbeeld

Raadpleeg het voorbeeld bij het commando DEFINE PAD.

Zie ook

DEFINE MENU, DEFINE PAD, DEFINE POPUP, ON SELECTION PAD

ON PAGE

Met het commando ON PAGE wordt een opgegeven instructie uitgevoerd wanneer een bepaalde regel op de actieve pagina is gepasseerd tijdens een ?/??- of EJECT PAGE-commando. ON PAGE wordt meestal bij het afdrukken gebruikt om pagina-afbrekingen met kop- en voetregels in een rapport te verwerken.

Syntaxis

ON PAGE [AT LINE <Nuitdr> <commando>]

Gebruik

Met ON PAGE wordt een opgegeven commando uitgevoerd wanneer in een ?/??- of EJECT PAGE-commando het regelnummer wordt bereikt dat is opgegeven in de AT LINE-clausule. De regelnummers worden tijdens het afdrukken bijgehouden met behulp van de systeemvariabele _plineno (raadpleeg hoofdstuk 5, "Systeemgeheugenvariabelen").

Met ON PAGE kunt u in een programma een geldig commando uitvoeren, zoals een routine voor kop- en voetregels. Het commando dat door ON PAGE wordt uitgevoerd wordt een *pagineerfunctie* genoemd.

Als u ON PAGE zonder argument opgeeft, wordt de pagineerfunctie uitgeschakeld.

Om de waarde van <Nuitdr> te bepalen voor de AT LINE-clausule, moet u de volgende formule gebruiken: <Nuitdr> = paginalengte - ondermarge - aantal voetregels.

De paginalengte is de waarde van de systeemgeheugenvariabele _plength. Het aantal voetregels is in de procedure voor de voetregels gedefinieerd (zie "Voorbeeld" hierna). De ondermarge is het aantal blanco regels onder de laatste regel van de voettekst.

dBASE IV houdt het aantal regels in de koptekst bij en stelt aan de hand daarvan het aantal regels op de betreffende pagina in. U moet zelf echter controleren of het aantal regels in de voettekst niet groter is dan het aantal resterende regels op de pagina. Als dat wel het geval is, loopt de voettekst door op de bovenzijde van de volgende pagina.

Als u de rapportgenerator hebt gebruikt om een rapport met kop- en voetregels te maken, wordt de pagineerfunctie tijdens het afdrukken automatisch geactiveerd met het commando REPORT FORM. U kunt de pagineerfunctie ook gebruiken voor andere afdrukfuncties, zoals DISPLAY of LIST.

Elke voettekstprocedure moet beginnen met een ?-commando om het regeldoorvoerteken te vervangen dat is vervallen doordat het commando ON PAGE is geactiveerd. U moet voettekstprocedures altijd beëindigen met EJECT PAGE. Procedures waarbij zones zijn betrokken, moeten met ?? worden beëindigd omdat met ? een wagenterugloop wordt gegeven vóór de volgende rapportzone, waardoor er een lege regel boven aan de pagina wordt afgedrukt.

Als het opgegeven <commando> een ander programma van dBASE of een door de gebruiker gedefinieerde functie moet uitvoeren, controleert u eerst of het betreffende programma of de betreffende functie al is gecompileerd. Tijdens het uitvoeren van een programma worden externe programmamodules niet gecompileerd.

Voorbeeld

Als u een boven- en ondermarge van 6 regels wilt hanteren, ziet een programma waarmee een voorraadoverzicht moet worden afgedrukt er ongeveer als volgt uit:

```
* Naam...: Voorraad.prg
ON PAGE AT LINE 60 DO Pag_onder
SET TALK OFF
SET PRINT ON
DO Kopreg                                   && Kopregel van eerste pagina afdrukken.
SCAN ALL
    ? ONDRDL_NR, OMSCHRIJV, VOORRAAD            && 3 velden afdrukken.
ENDSCAN
EJECT PAGE                                  && Pagineerfunctie activeren.
DO Voetreg                                  && Voetregel van laatste pagina afdrukken.
SET PRINT OFF
ON PAGE                                     && Pagineerfunctie uitschakelen.
SET TALK ON
RETURN
* EOP: Voorraad.prg
PROCEDURE Kopreg
EJECT PAGE                                  && Op nieuwe pagina beginnen.
?                                           && Kopregel afdrukken op regel twee.
? "Huidige voorraad" STYLE "B", DATE() AT 70
?
?                                           && Start LIST op de zevende regel.
RETURN
* EOP: Kopreg
PROCEDURE Voetreg
?
?                                           && Voetregel afdrukken op regel 63.
? "VERTROUWELIJK - niet bestemd voor anderen<" STYLE "B"
?? "Pagina" AT 70, LTRIM(STR(_pageno,4,0))
RETURN
* EOP: Voetreg
PROCEDURE Pag_einde
DO Voettkst                                 && Voettekst op actieve pagina afdrukken.
DO Koptekst                                 && Koptekst op actieve pagina afdrukken.
RETURN
* EOP: Pag_onder
```

Zie ook

?/??, FUNCTION, PRINTJOB, PROCEDURE, REPORT FORM, SET PRINTER, _plength, _plineno

ON READERROR

Met ON READERROR kunt u fouten opsporen en herstellen tijdens schermgrote bewerkingen.

Syntaxis

ON READERROR [<commando>]

Gebruik

Met ON READERROR kunt u na foutcontrole een commando, programma of procedure activeren. De fouten die met ON READERROR worden gevonden kunnen ongeldige datums zijn, een RANGE-specificatie die tijdens gegevensinvoer buiten het bereik valt, of een ongeldige VALID <voorwaarde>. Als deze fouten in uw programma worden gevonden, wordt het commando of programma uitgevoerd dat u bij ON READERROR hebt opgegeven. Als u ON READERROR zonder commando gebruikt, wordt de foutcontrole uitgeschakeld.

De commando's die niet zijn toegestaan in door de gebruiker gedefinieerde functies zijn ook niet toegestaan in het commando ON READERROR. Raadpleeg het gedeelte over het commando FUNCTION voor een lijst met deze commando's.

Tips

Dit commando is meestal een DO <commandobestand> om de fout te herstellen of hulpinformatie weer te geven. U kunt om de juiste invoer vragen door het programma geldige voorbeelden te laten geven.

Wanneer het opgegeven <commando> een ander dBASE-programma moet aanroepen, controleert u eerst of het programma al is gecompileerd. Tijdens de uitvoering van programma's worden externe programmamodules niet gecompileerd.

Zie ook

@, APPEND, BROWSE, CHANGE, EDIT, INSERT, ON ERROR, READ, SET FORMAT

ON SELECTION PAD

Met ON SELECTION PAD kunt u een bewerking combineren met een strip van een balkmenu. Als de opgegeven strip wordt geselecteerd, wordt het commando, de procedure of het programma in de commandoregel uitgevoerd.

Syntaxis

ON SELECTION PAD <stripnaam> OF <menunaam> [<commando>]

Gebruik

Wanneer u een willekeurige strip kiest uit een actief menu, wordt met ON SELECTION PAD het opgegeven <commando> uitgevoerd. De <commando>-clausule kan een commando van dBASE IV zijn, of de opdracht DO <programmanaam/procedurenaam>. In een procedure die wordt gecombineerd met ON SELECTION PAD kan gebruik worden gemaakt van functies als MENU() en PAD() om de naam van het actieve menu en de laatst gekozen strip te bepalen, en om de vereiste bewerkingen uit te voeren.

Wanneer een commando, procedure of programma is beëindigd, kan een nieuwe keuze worden gemaakt uit het menu.

Als u het commando ON SELECTION PAD zonder commando of programma geeft, wordt de opgegeven strip uitgeschakeld.

Met het commando ON SELECTION PAD kunnen andere commando's worden uitgevoerd, tenzij ON SELECTION PAD is geactiveerd in een door de gebruiker gedefinieerde functie of met het commando ON KEY, ON READERROR of ON PAGE.

Voorbeeld

In de voorbeeldbestanden wordt het menu Stop_pop geactiveerd als op de strip Stop van het balkmenu Hoofd op ↵ wordt gedrukt. Hiervoor wordt het volgende commando gebruikt:

```
. ON SELECTION PAD Stop OF Hoofd ACTIVATE POPUP Stop_pop
```

Zie ook

DEFINE MENU, DEFINE PAD, MENU(), ON PAD, PAD(), PROMPT()

ON SELECTION POPUP

Met ON SELECTION POPUP geeft u op welk commando of welke procedure moet worden uitgevoerd als een keuze wordt gemaakt in een pop-up menu.

Syntaxis

ON SELECTION POPUP <pop-up naam>/ALL [<commando>]

Gebruik

Geef dit commando om een van de aanwijzingen van het opgegeven pop-up menu te combineren met een procedure of een commando van dBASE IV. Gebruik het commando DO om een programma of procedure uit te voeren.

Als u ON SELECTION POPUP zonder commando gebruikt, wordt de optie die uit het pop-up menu is gekozen, niet uitgevoerd. U zou van deze mogelijkheid gebruik kunnen maken om een optie voor bepaalde gebruikers uit te schakelen. Als u in plaats van de pop-up naam ALL gebruikt, geldt dit commando voor alle pop-up menu's.

Door de gebruiker gedefinieerde menucommando's moeten in de volgende volgorde worden gegeven:

1. DEFINE POPUPS
2. ON SELECTION POPUP
3. ACTIVATE POPUP

Wanneer u een van de aanwijzingen van het pop-up menu kiest door op ↵ te drukken, wordt het menu inactief gemaakt totdat het commando of de procedure die bij ON SELECTION POPUP is opgegeven, is uitgevoerd.

OPMERKING
Nadat het commando ON SELECTION POPUP is gebruikt, wordt het actieve pop-up menu weer op het scherm afgebeeld wanneer het opnieuw wordt geactiveerd. Het scherm wordt echter niet opgeslagen voordat een van de ON-commando's wordt uitgevoerd. U moet het scherm via het programma opslaan en herstellen wanneer een van de ON-commando's wordt uitgevoerd.

Met het ON SELECTION POPUP-commando kunnen ook commando's worden uitgevoerd, tenzij ON SELECTION POPUP is geactiveerd in een door de gebruiker gedefinieerde functie of met het commando ON KEY, ON READERROR of ON PAGE.

Voorbeelden

Wanneer in het menu Stop_pop op ↵ wordt gedrukt, wordt de procedure Stop_pro uitgevoerd:

```
. ON SELECTION POPUP Stop_pop DO Stop_pro
```

Kenmerkend voor een procedure die wordt aangeroepen vanuit een menu, is dat de procedure zowel de DO CASE-structuur bevat om de keuze van de gebruiker te evalueren, als de bewerking die moet worden uitgevoerd naar aanleiding van de gemaakte keuze. De keuze van de gebruiker kan met de functies POPUP(), PROMPT() en BAR() worden bepaald.

Hierna volgt een voorbeeld van een procedure voor het menu Visie_pop:

```
DO CASE
    CASE BAR()=1
        APPEND BLANK
        EDIT NEXT 1
    CASE BAR() = 2
        EDIT NEXT 1
    CASE BAR() = 4
        DELETE
ENDCASE
RETURN
```

Zie ook

BAR(), DEFINE BAR, DEFINE POPUP, POPUP(), PROMPT()

PACK

Met PACK verwijdert u de records met een wismarkering uit het actieve database-bestand.

Syntaxis

PACK

Gebruik

Alle geopende indexbestanden worden automatisch opnieuw geïndexeerd.

Nadat een PACK-commando is gegeven, wordt de geheugenruimte die door de verwijderde records werd ingenomen, geschoond zodra het bestand wordt gesloten.

Denk eraan dat de commando's DIR en LIST/DISPLAY FILES de juiste omvang van de bestanden niet weergeven totdat de betreffende bestanden worden gesloten, als SET AUTOSAVE op OFF is ingesteld.

Speciale gevallen

In een netwerkomgeving moet het database-bestand in de modus EXCLUSIVE zijn geopend voordat u het commando PACK geeft.

Zie ook

COPY, DELETE, DELETED(), DIR, RECALL, REINDEX, SET AUTOSAVE, SET EXCLUSIVE, ZAP

PARAMETERS

Met het commando PARAMETERS worden lokale variabelenamen toegewezen aan de gegevens die afkomstig zijn uit het aanroepende programma. Bij dit commando kunt u variabelen opgeven die door het commando DO of DEBUG, of door een door de gebruiker gedefinieerde functie worden ingevoerd.

Syntaxis

PARAMETERS <parameterlijst>

Gebruik

In een programmabestand moet PARAMETERS het eerste uit te voeren commando zijn. In een procedurebestand is PROCEDURE <bestandsnaam> het eerste uit te voeren commando, direct gevolgd door het commando PARAMETERS. Als parameter kunt u elke geldige uitdrukking invoeren. De parameterlijst wijst lokale variabelenamen toe aan de parameters die afkomstig zijn uit de parameterlijst van het zendende programma. De lokale variabelen die met PARAMETERS worden gemaakt, worden overschreven wanneer het aanroepende programma weer is geactiveerd. De parameterlijst moet evenveel onderdelen bevatten als het aanroepende programma. U kunt maximaal 50 parameters door het aanroepende programma laten invoeren, maar hiervan mogen er niet meer dan tien constanten zijn en niet meer dan zes reekselementen. De overige parameters moeten variabelen zijn.

Als een geheugenvariabele als parameter wordt ingevoerd, mag de waarde van de betreffende variabele worden gewijzigd. De wijziging wordt overgebracht naar de variabele in het aanroepende programma. Als een uitdrukking als parameter wordt ingevoerd, wordt de waarde van de betreffende uitdrukking opgeslagen in een geheugenvariabele van het ontvangende programma.

Als u een reekselement als parameter in een subroutine invoert (het element wordt op basis van de waarde ingevoerd, niet op basis van de verwijzing), heeft de wijziging alleen invloed op die routine. De oorspronkelijke waarde blijft gelijk wanneer u met RETURN naar de aanroepende routine terugkeert. U kunt de waarde van een reekselement wijzigen door de reeksnaam als geheugenvariabele te gebruiken in plaats van de naam als parameter in te voeren.

Zie ook

@, DO, FUNCTION, PROCEDURE, SET PROCEDURE, STORE

PLAY MACRO

Met dit commando kunt u macro's uit de actieve macrobibliotheek uitvoeren.

Syntaxis

PLAY MACRO <macronaam>

Gebruik

Een macro is een reeks toetsaanslagen die wordt opgeslagen in de actieve macrobibliotheek. U geeft elke macro een unieke label en u kunt de reeks toetsaanslagen ieder moment uitvoeren door de label op te roepen.

Wanneer u een macro maakt vanuit het Control Center, wijst u een toets en een naam aan de macro toe. Macro's die buiten het Control Center worden gemaakt, worden geïdentificeerd aan de hand van de bijbehorende macrotoets.

De macrotoets kan **Alt-F1** tot en met **Alt-F9** zijn, of **Alt-F10** gevolgd door een van de letters A tot en met Z. U kunt dus maximaal 35 toetsen toewijzen en maximaal 35 unieke macro's tegelijk maken. Een macronaam mag 10 tekens lang zijn en mag geen spaties bevatten.

De macro's die u maakt en opslaat kunnen op drie manieren worden uitgevoerd:

- door op **F10** te drukken en daarna **Uitvoeren** te kiezen uit het menu **Diversen** van het Control Center;
- door op de macrotoets op het toetsenbord te drukken. Dit kunnen de toetsen **Alt-F1** tot en met **Alt-F9** zijn, of **Alt-F10** gevolgd door een van de letters A tot en met Z (hoofdletters of kleine letters);
- met het commando PLAY MACRO gevolgd door de macronaam.

Omdat u met PLAY MACRO een macro kunt uitvoeren aan de hand van de macronaam, kunt u dit commando in programmabestanden opnemen om een macro uit te voeren zonder de gebruiker te vragen op de macrotoets te drukken.

PLAY MACRO heeft twee belangrijke eigenschappen wanneer het binnen een programma wordt gebruikt. Ten eerste wordt de macro in het geheugen opgeslagen totdat uitvoer kan worden gesimuleerd. Als u PLAY MACRO vanaf de commandostip opgeeft, heeft dat direct resultaat omdat de commandostip altijd gereed is voor invoer, of deze nu rechtstreeks wordt getypt of in een macro wordt uitgevoerd. In een programma worden de toetsaanslagen niet uitgevoerd totdat de betreffende toetsaanslagen als invoer moeten worden gebruikt in een ander commando. Dat gebeurt bijvoorbeeld als u eerst het commando PLAY MACRO geeft en daarna APPEND, zodat de toetsaanslagen bij de uitvoering in een record worden opgeslagen.

In een programma wordt een reeks PLAY MACRO-commando's in het geheugen geplaatst op een "last in, first out"-basis (LIFO). Als u bijvoorbeeld een programma schrijft waarin achtereenvolgens drie PLAY MACRO-commando's worden gegeven, wordt de derde macro het eerst uitgevoerd en de eerste macro het laatst.

Zie ook

KEYBOARD, RESTORE MACROS, SAVE MACROS

PRINTJOB/ENDPRINTJOB

PRINTJOB/ENDPRINTJOB zijn gestructureerde programmeercommando's waarmee afdrukopdrachten worden bestuurd.

Syntaxis

PRINTJOB <commando's> ENDPRINTJOB

Gebruik

Met PRINTJOB activeert u de instellingen van systeemgeheugenvariabelen voor een afdrukopdracht.

Een systeemgeheugenvariabele is een variabele die wordt geïnitieerd door dBASE IV en die u kunt wijzigen. Raadpleeg hoofdstuk 5, "Systeemgeheugenvariabelen", voor een bespreking van alle systeemgeheugenvariabelen.

Met het commando PRINTJOB worden de volgende instructies uitgevoerd:

- De codes voor het starten van de afdrukopdracht die worden gedefinieerd in de systeemvariabele _pscodes, verzenden.
- Het papier doorvoeren als _peject is ingesteld op "BEFORE" of "BOTH".
- _pcolno instellen op 0

Met het commando ENDPRINTJOB worden de volgende instructies uitgevoerd:

- De codes voor het beëindigen van de afdrukopdracht die worden gedefinieerd in de systeemvariabele _pecodes, verzenden.
- Het papier doorvoeren als _peject is ingesteld op "AFTER" of "BOTH".
- De inhoud van de afdrukbuffer het aantal malen afdrukken dat met _pcopies is gedefinieerd. Als _pcopies groter is dan 1, wordt de uitvoer als spool-bestand naar de vaste schijf overgebracht en wordt dit daarna het aantal malen afgedrukt dat met _pcopies is opgegeven.

In uw code moeten de definities van deze systeemvariabelen voor het commando PRINTJOB staan. Indien nodig moet u de systeemvariabelen die in de code zijn gebruikt na het commando ENDPRINTJOB opnieuw definiëren.

U kunt ook andere systeemvariabelen definiëren voor een bepaalde afdrukopdracht. Als u een bepaalde functie voor de gehele afdrukopdracht wilt uitvoeren, definieert u de variabele voor het commando PRINTJOB. U kunt bijvoorbeeld het afdrukken na iedere pagina stoppen of een bepaalde printermarge instellen.

U kunt PRINTJOB en ENDPRINTJOB alleen gebruiken in een programma. Afdrukopdrachten kunnen niet worden genest. Als u met PRINTJOB afdrukt, kunt u de reeksnummers van de eerste en de laatste pagina met de systeemgeheugenvariabelen _pbpage en _pepage instellen.

Voorbeelden

In het volgende voorbeeld wordt een aangepaste afdrukopdracht samengesteld met het database-bestand Afnemers.dbf.

```
* Klant.prg
USE Afnemers
_peject = "AFTER"
SET PRINT ON
PRINTJOB
ON PAGE AT LINE _pLength-1 DO PagOnder
SCAN
    ?? Klant_nr AT 0,Klantnaam AT 8
    ?
ENDSCAN
ENDPRINTJOB
SET PRINT OFF
RETURN
* EOP: Klant.prg

PROCEDURE PagOnder
EJECT PAGE
?? DATE() AT 1, "Pagina:" AT 67, _pageno PICTURE "999" AT 73
?
?
RETURN
* EOP: PagOnder
```

In het volgende programma worden drie kopieën van een rapport afgedrukt, vanaf pagina tien tot het einde, en wordt na het afdrukken een pagina doorgevoerd. Het commando REPORT FORM dat het rapport uitvoert, bevat de PRINTJOB-instellingen zodat u de commando's PRINTJOB/ENDPRINTJOB niet hoeft te gebruiken als u REPORT FORM gebruikt. Deze routine stelt _pbpage en _pcopies opnieuw in op de standaardwaarde als de afdrukopdracht is uitgevoerd.

```
_pbpage = 10
_peject = "AFTER"
_pcopies = 3
REPORT FORM Rekening TO PRINT
_pbpage = 1
_pcopies = 1
```

Zie ook

ON PAGE, PROCEDURE, REPORT FORM, RETURN, SET PRINTER, SET TALK, _pcopies, _pecodes, _peject, _plineno, _pscodes

PRIVATE

Met PRIVATE kunt u in een programma van een lager niveau lokale geheugenvariabelen maken met dezelfde naam als de geheugenvariabelen die zijn gemaakt in een aanroepend programma of die eerder als PUBLIC zijn gedefinieerd.

Syntaxis

PRIVATE ALL [LIKE/EXCEPT <structuur>]

of

PRIVATE <gehvarlijst>

Gebruik

Wijzigingen in geheugenvariabelen die PRIVATE zijn hebben geen gevolgen voor geheugenvariabelen met dezelfde naam in een ander programma. Oorspronkelijke waarden van PRIVATE geheugenvariabelen worden in het geheugen opgeslagen met de namen van de programma's waarin deze zijn gemaakt.

Wanneer het programma met de PRIVATE geheugenvariabelen wordt beëindigd, worden de waarden die door PRIVATE zijn verborgen, opnieuw ingesteld. De PRIVATE-waarden zijn niet meer geldig.

Wanneer u een private-systeemgeheugenvariabele in een dBASE IV-procedure definieert, wordt een PRIVATE kopie gemaakt waaraan de actieve waarde voor de betreffende systeemvariabele wordt toegewezen. Wanneer u de procedure verlaat, wordt de oorspronkelijke waarde van de systeemgeheugenvariabele opnieuw bepaald en opnieuw ingesteld.

Zie ook

DECLARE, DO, PARAMETERS, PUBLIC, STORE

PROCEDURE

Met het commando PROCEDURE geeft u het begin van een subroutine aan.

Syntaxis

PROCEDURE <naam procedure>

Gebruik

Een procedure is een subroutine die u kunt uitvoeren met het commando DO<procedure>. U kunt procedures opnemen in een programmabestand of opslaan in een apart procedurebestand. De procedure wordt in een aantal bestanden in de volgende volgorde opgezocht:

1. Het actieve objectbestand.
2. Het bestand dat is opgegeven met SET PROCEDURE, waarin de bestandsnaam van de PROCEDURE wordt vastgesteld en de eerste PROCEDURE met die naam die wordt aangetroffen, wordt gebruikt.
3. Andere geopende objectbestanden, te beginnen met het laatst geopende.
4. Een objectbestand met de procedurenaam, waarna het bestand wordt geopend.
5. Een programmabestand met de procedurenaam, waarna het bestand wordt gecompileerd.
6. Een SQL-programmabestand met de procedurenaam, waarna het bestand wordt gecompileerd.

Procedurenamen kunnen worden herhaald, omdat door het bovenstaande zoekpatroon een procedure verborgen blijft wanneer een ander bestand een hogere plaats inneemt in de zoekvolgorde.

Het aantal procedures dat u in een programma- of procedurebestand kunt opnemen, is vooral afhankelijk van de beschikbare hoeveelheid werkgeheugen, met een maximum van 963 procedures per bestand. Het begin van elke procedure wordt aangegeven met het sleutelwoord PROCEDURE en het einde wordt aangegeven met RETURN. Na het commando RETURN, aan het einde van de procedure, kunt u geen uitdrukking opgeven.

Procedurenamen kunnen negen tekens lang zijn. De naam mag bestaan uit letters, cijfers en onderstrepingstekens en moet beginnen met een letter of een cijfer. U kunt een procedure niet dezelfde naam geven als het programmabestand waarin de procedure is opgenomen.

Richtlijnen voor het programmeren

Aan het begin van elk objectbestand (.dbo) van een programma wordt een procedurelijst bijgehouden. Het programma zelf wordt ook als een procedure beschouwd en krijgt de naam van het bronbestand als procedurenaam. Deze procedurenaam staat als eerste in de procedurelijst. Elke daaropvolgende procedure, of deze nu in het actieve bronbestand of in een apart procedurebestand staat, wordt toegevoegd aan de lijst wanneer het bronbestand wordt gecompileerd.

Neem bijvoorbeeld het volgende programma, Hoofd:

```
<EXAMPLE>*Hoofd.prg
!commando's!
DO A
DO B
DO C
RETURN

PROCEDURE A
!commando's!
RETURN

PROCEDURE B
!commando's!
RETURN

PROCEDURE C
!commando's!
RETURN
```

Er worden vier procedures opgenomen in de procedurelijst van het gecompileerde objectbestand: Hoofd (de standaardprocedurenaam), A, B en C. De standaardprocedurenaam wordt alleen toegekend aan de code aan het begin van een programmabestand.

Als u de procedures in één programmabestand plaatst, voorkomt u dat er veel verschillende programmabestanden worden gebruikt. Programma's worden dan sneller uitgevoerd, doordat dBASE IV de desbetreffende programmabestanden niet eerst hoeft te openen en te sluiten. U kunt ook het commando SET PROCEDURE TO <naam procedure> geven voor procedures die moeten worden aangeroepen met het commando DO.

Elke procedure in een actief objectbestand kan worden aangeroepen met DO <naam procedure>. Wanneer B.dbo wordt opgeroepen door A.dbo en C.dbo wordt opgeroepen door B.dbo, staan alle procedures in A, B en C ter beschikking van de procedures in C.

Voorbeelden

Het volgende bestand bevat twee routines waarmee meldingen op het scherm worden weergegeven:

```
* Dit is een voorbeeld van het procedurebestand Proc1.prg

PROCEDURE Melding1
    @ 15,0 CLEAR
    @ 18,0 SAY "Saldotekort geconstateerd"
RETURN

PROCEDURE Melding2
    @ 10,0 CLEAR
    @ 15,10 SAY "Geen betaling ontvangen"
    @ 17,10 SAY "Rekening blokkeren s.v.p."
RETURN
* EOP: Proc1.prg
```

PROCEDURE

Een procedure in Proc1.prg wordt als volgt geactiveerd:

```
. SET PROCEDURE TO Proc1
. DO Melding1
```

Zie ook

COMPILE, DEBUG, DO, FUNCTION, PARAMETERS, RETRY, RETURN, SET PROCEDURE

PROTECT

Met het commando PROTECT activeert en onderhoudt u de beveiliging van een dBASE IV-systeem.

Syntaxis

PROTECT

Gebruik

Dit menugestuurde commando wordt binnen dBASE IV gegeven door de database-beheerder, die verantwoordelijk is voor gegevensbeveiliging. PROTECT werkt zowel in een zelfstandig systeem als in een netwerk.

PROTECT is optioneel. Als u uw gegevens beveiligt, controleert het beveiligingssysteem altijd of gebruikers toegang hebben tot de database-bestanden die zij proberen te openen.

Dit commando resulteert in een schermgroot menu. De eerste keer dat u het commando PROTECT geeft, wordt u gevraagd een wachtwoord voor de beheerder in te voeren en te bevestigen.

WAARSCHUWING
Onthoud het beheerderswachtwoord goed. Alleen met dit wachtwoord hebt u toegang tot het beveiligingssysteem. Wanneer het beveiligingssysteem eenmaal is ingesteld, kunt u de beveiliging alleen wijzigen als u het juiste wachtwoord invoert nadat u het commando PROTECT hebt gegeven. Schrijf het wachtwoord van de beheerder op en bewaar het op een veilige plaats. U kunt het wachtwoord op geen en*kele manier opvragen.*

Als u van plan bent SQL te gaan gebruiken, is het van belang dat u ook de aanmeldingsnaam SQLDBA opgeeft. Dit is de aanmeldingsnaam van de "super user". Onder deze naam kunt u alle bewerkingen in SQL-modus uitvoeren. Het toegangsrecht voor velden en bestanden wordt bepaald door de SQL-commando's GRANT en REVOKE op basis van de aanmeldingsnamen die zijn toegekend met PROTECT.

PROTECT kent drie verschillende soorten database-beveiliging:

- *Aanmeldingsbeveiliging*, waardoor alleen daartoe gemachtigd personeel toegang heeft tot dBASE IV
- *Toegangsbeveiliging voor bestanden en velden*, waardoor u kunt bepalen tot welke bestanden en velden een gebruiker toegang heeft
- *Gegevenscodering*, waardoor dBASE-bestanden zodanig worden gecodeerd dat alleen daartoe gemachtigde gebruikers deze kunnen lezen.

Tabel 2-6 geeft een overzicht van de verschillende soorten database-beveiliging, van de methoden en van het resultaat van de beveiliging.

Tabel 2-9 dBASE-beveiliging

Type beveiliging	Te definiëren:	Resultaat:
Aanmelding	Naam en wachtwoord van gebruiker	Invloed op toegang tot dBASE IV
Toegang tot bestand en veld.	Toegangniveaus	.Invloed op de toegang tot gegevensbestanden, velden in gegevensbestanden en toepassingscode
Gegevenscodering	Gebruikers- en bestandsgroep	Automatische codering en decodering van gegevens

U hoeft niet alle drie de niveaus te installeren. Desgewenst kunt u stoppen op het aanmeldingsniveau. De verschillende soorten beveiliging moeten worden geïnstalleerd in de volgorde waarin deze in tabel 2-9 worden opgesomd.

Aanmeldingsbeveiliging is het eerste beveiligingsniveau. Wanneer een beveiligingssysteem eenmaal is geïnstalleerd, hebben gebruikers alleen toegang tot dBASE IV nadat zij een geldige aanmeldingsnaam hebben opgegeven.

Toegangsbewaking is het volgende beveiligingsniveau. Met toegangsbewaking worden beperkingen gesteld aan hetgeen een gebruiker kan doen met een database-bestand en met de gegevens in het bestand. Ook kan op deze wijze de verwerking van toepassingscode worden gestuurd. De *toegangsniveaus* lopen van 1 tot en met 8, waarbij 1 het hoogste en 8 het laagste toegangsrecht is. U stelt een toegangsniveau in voor elke gebruiker in het gebruikersprofiel en ook toegangsniveaus voor bestanden en velden in het *toegangsrechtenoverzicht*.

U stelt toegangsrechten in voor een database-bestand door een willekeurige combinatie van toegansniveaus toe te kennen voor lezen, bijwerken, uitbreiden of wissen.

Met gegevenscodering wordt de database zodanig vervormd dat alleen gebruikers met de vereiste toegangsrechten de informatie in het bestand kunnen lezen.

Het bestand Dbsystem.db

Met PROTECT maakt en onderhoudt u een systeembestand voor de wachtwoorden, Dbsystem.db, waarin een record staat voor iedere gebruiker die toegang heeft tot het beveiligde systeem. Een dergelijk record wordt een *gebruikersprofiel* genoemd. In het record staan de aanmeldingsnaam van de gebruiker, de volledige naam, het wachtwoord, de groepsnaam en het toegangsniveau. Wanneer een gebruiker het commando dBASE invoert op een netwerkstation, wordt het bestand Dbsystem.db gezocht. Als het bestand is gevonden, wordt begonnen met het aanmeldingsproces. Zo niet, dan volgt er geen aanmeldingsproces.

Dbsystem.db is een gecodeerd bestand. Het is verstandig een afdruk te maken van een deel van de informatie of van alle informatie in Dbsystem.db. Wanneer bijvoorbeeld een gebruiker een aanmeldingswaarde is vergeten (zoals groepsnaam, aanmeldingsnaam of wachtwoord), is het handig om die informatie bij de hand te hebben.

Met het menu **Rapport** kunt u beveiligingsinformatie over gebruikers en bestanden weergeven of afdrukken.

Bestandstoegangsrechten

U kunt de toegangsrechten opgeven voor maximaal acht database-bestanden tegelijk. Wanneer u probeert de toegangsrechten tot een negende bestand in te stellen, verschijnt de foutmelding **Te veel bestanden geopend**. Wanneer u op deze wijze acht overzichten van bestandstoegangsrechten hebt gemaakt, moet u deze acht overzichten opslaan voordat u een nieuw overzicht kunt maken.

Bestandstoegangsrechten hebben een lagere prioriteit dan de schrijfbeveiliging van een bestand op het niveau van het besturingssysteem.

Het wijzigen van bestandstoegangsrechten

Wanneer u een bestand probeert te openen, wordt gecontroleerd of er voor het bestand al toegangsrechten zijn vastgesteld. Als dat het geval is, worden de desbetreffende waarden weergegeven in het menu. U kunt deze waarden dan wijzigen. Als het toegangsrechtenoverzicht al eens is opgeslagen, verschijnt de melding **<bestandsnaam>.crp bestaat al. Overschrijven? (J/N)**. Druk op J om de waarden te wijzigen. (Deze melding is niet afhankelijk van de instelling van SET SAFETY.)

Nadat u wijzigingen hebt aangebracht, slaat u het toegangsrechtenoverzicht op.

PROTECT verlaten

In het menu **Stop** staan drie opties:

Kies **Opslaan** om alle nieuwe en bijgewerkte gebruikersprofielen en overzichten van bestandstoegangsrechten die zijn gemaakt tijdens de PROTECT-sessie, op te slaan. Gebruikersprofielen worden opgeslagen in het actieve Dbsystem.db-bestand. Overzichten van bestandstoegangsrechten worden vastgelegd in de bestandsstructuur van de database. U kunt gebruikersprofielen op ieder gewenst moment tijdens een PROTECT-sessie opslaan. Overzichten van bestandstoegangsrechten moeten worden weggeschreven nadat u er acht hebt gedefinieerd of gewijzigd. Een database-bestand wordt gecodeerd wanneer er voor het betrokkken bestand een toegangsrechtenoverzicht wordt opgeslagen.

Kies **Annu**leren om alle gebruikersprofielen en overzichten van bestandstoegangsrechten die tijdens de PROTECT-sessie zijn gemaakt of bewerkt, maar nog niet zijn weggeschreven, te laten vervallen.

Kies **Einde** om de PROTECT-sessie te beëindigen. Wanneer nieuwe en bijgewerkte gebruikersprofielen en bijgewerkte overzichten van bestandstoegangsrechten nog niet zijn gecodeerd en opgeslagen, wordt dit alsnog gedaan.

Gegevenscodering

Als een database-systeem is beveiligd, worden de database-bestanden en de bijbehorende indexen en memobestanden automatisch door dBASE IV gecodeerd en gedecodeerd.

Als u het overzicht van de bestandstoegangsrechten van een database opslaat, wordt er een

gecodeerde versie van het database-bestand gemaakt met de extensie .crp. Het is raadzaam het ongecodeerde bestand naar een diskette te kopiëren en de diskette op een veilige plaats te bewaren. Het commando MODIFY STRUCTURE heeft namelijk alleen effect als het database-bestand niet is gecodeerd.

Op de volgende wijze activeert u de beveiliging:

1. Kopieer het gecodeerde bestand (.crp) over het ongecodeerde bestand (.dbf) en wijzig de extensie in .dbf:

```
. <COPY FILE Bestnaam.crp TO Bestnaam.dbf
```

2. Verwijder het .crp-bestand:

```
. ERASE Bestnaam.crp
```

3. Wijzig de extensie van het .cpt-bestand in .dbt.

Indexbestanden worden alleen gecodeerd als het database-bestand is gecodeerd op het moment dat u het bestand indexeert of opnieuw indexeert.

Met het commando SET ENCRYPTION bepaalt u of gekopieerde bestanden worden gecodeerd.

Zie ook

LOGOUT, SET ENCRYPTION

Als u dBASE IV hebt aangeschaft, vindt u in hoofdstuk 14, "De optie Diversen", in *Menu's van dBASE IV* meer informatie over PROTECT.

Als u de Developer's Edition van dBASE IV hebt aangeschaft, vindt u meer informatie in hoofdstuk 5, "Beveiliging van dBASE", in *dBASE IV in netwerken.*

PUBLIC

Met het commando PUBLIC kunt u geheugenvariabelen en reekselementen opgeven die kunnen worden gebruikt en gewijzigd in elk programma of subprogramma van dBASE. In tegenstelling tot PRIVATE geheugenvariabelen en reekselementen, vervallen PUBLIC geheugenvariabelen niet nadat het programma is uitgevoerd.

Syntaxis

PUBLIC <lijst geheugenvariabelen>/ ARRAY <reeksnaam 1> [{<aantal rijen>,}
<aantal kolommen>] {,<reeksnaam 2>[{<aantal rijen>,}
<aantal kolommen>]...}

In de syntaxis duiden accolades optionele elementen aan. De rechte haken zijn een verplicht onderdeel van de syntaxis van het commando PUBLIC.

Gebruik

PUBLIC geheugenvariabelen of reekselementen zijn globale variabelen die u in ieder programma kunt gebruiken. Geheugenvariabelen die u globaal wilt maken, moeten PUBLIC worden gemaakt voordat er een waarde aan wordt toegekend. Een geheugenvariabele die wordt opgegeven bij de commandostip is PUBLIC. Systeemgeheugenvariabelen zijn altijd PUBLIC. De waarde van systeemvariabelen kan in ieder programma worden gewijzigd.

Als een programma of procedure wordt aangeroepen, kunt u de waarde van een PUBLIC geheugenvariabele tijdelijk verbergen door PRIVATE geheugenvariabelen met dezelfde naam op te geven of door een parameter met dezelfde naam in een procedure op te nemen.

Variabelen die zijn opgeslagen in een geheugenbestand worden altijd teruggezet als PUBLIC variabelen als het commando RESTORE wordt gegeven vanaf de commandostip. Variabelen die in een programma van dBASE worden teruggezet, moeten eerst opnieuw worden gedefinieerd als PUBLIC variabelen. Met het commando RESTORE FROM <bestandsnaam> ADDITIVE kunt u de variabelen dan terugzetten als PUBLIC variabelen. Bestaande variabelen met dezelfde naam worden daarbij overschreven.

OPMERKING
PUBLIC variabelen zijn logische variabelen zolang u deze niet hebt geïnitialiseerd.

PUBLIC

Voorbeeld

Met de volgende commando's worden in een programmabestand van dBASE de geheugenvariabelen Pagina en Antwoord PUBLIC gemaakt en geïnitialiseerd:

```
*Hoofd.prg
DO Init_var
RETURN
PROCEDURE Init_var
PUBLIC Pagina, Antwoord
Pagina = 1
Antwoord = "J"
RETURN
```

Binnen een dBASE-programmabestand worden met het volgende commando twee public reeksen gemaakt met de namen Deel en Artikelen:

```
PUBLIC ARRAY Deel[6,3], Artikelen[8,3]
```

Zie ook

DECLARE, DO, PARAMETERS, PRIVATE, RELEASE, RESTORE, SAVE, STORE

QUIT

Met QUIT sluit u alle geopende bestanden, beëindigt u dBASE IV en keert u terug naar DOS.

Syntaxis

QUIT

Gebruik

Dit is de enige veilige methode om dBASE IV te verlaten en terug te keren naar DOS. Als u dBASE IV verlaat door de computer opnieuw op te starten, kunnen geopende bestanden worden beschadigd en kunnen gegevens verloren gaan.

READ

Met READ activeert u alle @...GET-opdrachten die zijn gegeven na de laatste keer dat het commando CLEAR, CLEAR ALL, CLEAR GETS of READ is gegeven. READ wordt vooral toegepast in programmabestanden, en dient voor het invoeren of bewerken van gegevens op het hele beeldscherm.

Syntaxis

READ [SAVE]

Gebruik

Met dit commando kunt u door de gebruiker ingevoerde gegevens opslaan in geheugenvariabelen of in velden. Door het commando in combinatie met @...GET te gebruiken, slaat u de ingevoerde gegevens op als geheugenvariabele. Wanneer u de gebruiker meerdere gegevens wilt laten invoeren, in plaats van één gegeven tegelijk (zoals bij ACCEPT, INPUT of WAIT), geeft u een aantal @...GET-opdrachten, gevolgd door één READ.

Bij het commando READ worden dezelfde bewerktoetsen gebruikt als bij de commando's APPEND en EDIT. Als u het commando READ hebt gegeven, kunt u geheugenvariabelen bewerken met het @-commando.

Nadat u de gewenste gegevens hebt ingevoerd, wordt de informatie uit alle GET-opdrachten gewist, tenzij u de optie SAVE gebruikt.

Wanneer er een indelingsbestand geopend is, worden door READ eerst de @...GET-opdrachten op het scherm afgebeeld en vervolgens het indelingsbestand geactiveerd en het scherm overschreven. Zorg ervoor dat de indelingsbestanden worden gesloten nadat ze zijn bewerkt, zodat ze niet door het commando READ worden geactiveerd en het scherm niet wordt overschreven; overtuig u er anders van dat alle @...GET-opdrachten in het indelingsbestand zijn opgenomen. De indeling heeft prioriteit boven de besturing van het scherm.

Optie

Wanneer u het commando READ SAVE geeft, worden de GET-opdrachten niet gewist. In dat geval worden, de volgende keer dat u het commando READ geeft, dezelfde gegevens gebruikt. Als u de optie SAVE opneemt in het commando, moet u eerst het commando CLEAR GETS geven voordat u een nieuwe verzameling GET-opdrachten opgeeft.

Richtlijnen voor het programmeren

Als u gebruik wilt maken van een indelingsbestand (.fmt) dat uit meerdere pagina's bestaat en waarin de @...SAY...GET-opdrachten zijn verspreid over 2 tot 32 beeldschermpagina's, geeft u met het commando READ aan, waar u een nieuwe pagina wilt laten beginnen. Wanneer met CREATE/MODIFY SCREEN gegenereerde indelingsbestanden meerdere pagina's lang zijn, worden automatisch READ-commando's geplaatst.

Voorbeeld

Met het commando READ kunt u de inhoud van een geheugenvariabele bewerken:

```
. STORE SPACE(25) TO Mnaam
. @ 10,10 SAY "Geef uw naam op: " GET Mnaam
. READ
```

De cursor wordt op de variabele Mname geplaatst, zodat u de variabele kunt bewerken. Met de functie SPACE(25) wordt een lege reeks van 25 tekens geïnitialiseerd.

Zie ook

@, CLEAR, CLEAR GETS, CLEAR MEMORY, CREATE/MODIFY SCREEN, ON ESCAPE, REPLACE, SET DEVICE, SET FORMAT, STORE

RECALL

Met het commando RECALL herstelt u records in het actieve database-bestand, die van een wismarkering zijn voorzien.

Syntaxis

RECALL [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstelling

Alleen de wismarkering van het actieve record wordt verwijderd, tenzij anders wordt bepaald door het <bereik> of de FOR- of WHILE-clausule.

Speciale gevallen

Records die uit de database zijn verwijderd met de commando's PACK of ZAP, kunnen niet worden niet teruggehaald met RECALL.

Als SET DELETED op ON is ingesteld, verplaatst u de recordwijzer met behulp van GOTO RECORD <n> naar het record dat moet worden teruggehaald. U kunt ook het nummer van het record opgeven met RECALL RECORD <n>. Wanneer SET DELETED op ON is ingesteld, wordt met RECALL ALL geen enkel record teruggehaald.

Voorbeelden

Op de volgende wijze herstelt u alleen het eerste record van de database, terwijl de records 1, 5 en 10 van een wismarkering zijn voorzien.

```
. USE Magazijn
. RECALL
    1 record hersteld
```

Als u record 10 wilt herstellen, gaat u als volgt te werk:

```
. RECALL RECORD 10
    1 record hersteld
```

Als u record 5 en alle andere records met een wismarkering wilt herstellen, doet u dat als volgt:

```
. RECALL ALL
    17 records hersteld
```

Zie ook

DELETE, DELETED(), PACK, SET DELETED, ZAP

REINDEX

Met het commando REINDEX kunt u alle actieve indexbestanden (.ndx) en meervoudige indexbestanden (.mdx) in het actieve werkgebied bijwerken, inclusief het bijbehorende .mdx-bestand.

Syntaxis

REINDEX

Gebruik

Met dit commando indexeert u alle geopende .ndx- en .mdx-bestanden in het actieve werkgebied, inclusief de labels in .mdx-bestanden, opnieuw. Wanneer u een bestand opnieuw indexeert, dat is gemaakt terwijl SET UNIQUE was ingesteld op ON of waarbij UNIQUE in de syntaxis van INDEX was opgenomen, behoudt het indexbestand de status UNIQUE, ongeacht of UNIQUE op dat moment op ON of OFF is ingesteld.

In een netwerk moet het database-bestand in exclusief gebruik zijn, voordat u .mdx-bestanden opnieuw kunt indexeren.

Voorbeeld

```
. USE Transact
. REINDEX
Index wordt opnieuw samengesteld - C:\DBASE\TRANSACT.MDX Tag: KLANT_NR
    100 % geïndexeerd                    12 records geïndexeerd
Index wordt opnieuw samengesteld - C:\DBASE\TRANSACT.MDX Tag: BESTEL_NR
    100 % geïndexeerd                    12 records geïndexeerd
```

Zie ook

NDEX, KEY(), MDX(), NDX(), PACK, SET INDEX, SET ORDER,

SET UNIQUE, USE, ZAP

RELEASE

Met het commando RELEASE verwijdert u geheugenvariabelen en maakt u dus geheugen vrij voor andere toepassingen. RELEASE kan in combinatie met het juiste sleutelwoord ook worden gebruikt om met LOAD geladen assembleertaalprogramma's, menu's, pop-ups, schermen en vensters uit het geheugen te verwijderen.

Syntaxis

RELEASE <lijst geheugenvariabelen>

of

RELEASE ALL [LIKE/EXCEPT <structuur>]

of

RELEASE MODULE [<namenlijst modules>]
    /MENUS [<namenlijst menu's>]
    /POPUPS [<namenlijst pop-ups>]
    /SCREENS [<namenlijst schermen>]
    /WINDOWS [<namenlijst vensters>]

Gebruik

In de interactieve modus worden met RELEASE ALL alle geheugenvariabelen behalve de systeemgeheugenvariabelen verwijderd. In een programma worden met RELEASE ALL alleen PRIVATE geheugenvariabelen verwijderd die zijn gemaakt in het actieve programma of in een programma op een lager niveau. PRIVATE variabelen die in een programma op een hoger niveau zijn gemaakt worden niet verwijderd, behalve als het commando bij de commandostip wordt gegeven. In een programma worden met RELEASE ALL geen PUBLIC variabelen verwijderd.

Opties

<lijst geheugenvariabelen> is een lijst met namen, gescheiden door komma's.

Met RELEASE MODULE verwijdert u een met LOAD geladen module uit het geheugen. Wanneer u een modulenaam opgeeft, moet u niet de extensie .bin opnemen.

Met RELEASE MENUS kunt u de opgegeven menu's van het scherm en uit het geheugen verwijderen. Alle ON SELECTION- en ON PAD-commando's die aan die menu's zijn gekoppeld, worden ook verwijderd. Een menu kan niet worden verwijderd als het op dat moment actief is, maar wel zodra het inactief is gemaakt. Als u geen lijst met menu's opgeeft, worden alle menu's verwijderd.

Met RELEASE POPUPS kunt u de opgegeven pop-up menu's van het scherm en uit het geheugen verwijderen. Als u het actieve pop-up menu opgeeft in de namenlijst, wordt het menu inactief gemaakt. De ON SELECTION POPUP-commando's, die zijn gekoppeld aan de pop-up menu's in de namenlijst, worden gewist voordat de pop-up menu's worden verwijderd. Als u geen namenlijst opgeeft, worden alle pop-up menu's van het scherm en uit het geheugen verwijderd.

Met RELEASE SCREENS kunt u de opgeslagen schermafbeeldingen uit het geheugen verwijderen. In de schermnamenlijst staan de namen van eerder opgeslagen schermen. Wanneer geen namen worden uitgelijst, verwijdert u met RELEASE SCREENS alle opgeslagen schermen.

Met RELEASE WINDOWS kunt u de opgegeven vensters van het scherm en uit het geheugen verwijderen. In de vensternamenlijst staan de namen van eerder gedefinieerde vensters, gescheiden door komma's. Tekst die werd bedekt door de verwijderde vensters, verschijnt weer op het scherm. Met dit commando kunt u selectief vensters uit het geheugen verwijderen.

Voorbeelden

Op de volgende wijze kunt u alle geheugenvariabelen die beginnen met de letter *m*, opheffen:

```
. RESTORE FROM Mem
. RELEASE ALL LIKE m*
```

Op de volgende manier worden alle geheugenvariabelen opgeheven, behalve de geheugenvariabelen waarvan de letter <CUR>x<SIEF> het derde teken is:

```
. RESTORE FROM Mem
. RELEASE ALL EXCEPT ??x*
```

U kunt de variabelen Gehvar, Pagina's, Pgx1 en Pgx2 als volgt opheffen:

```
. RESTORE FROM Mem
. RELEASE Gehvar, Pagina's, Pgx1, Pgx2
```

In het volgende voorbeeld wordt een verzameling pop-ups opgeheven die behoort bij een menu in een financieel programma:

```
. RELEASE POPUPS Krediet, Ouderdom, Debitren
```

Zie ook

CALL, CALL(), CLEAR ALL, CLEAR MEMORY, DEFINE MEMO, DEFINE POPUP, DEFINE WINDOW, LOAD, ON PAD, ON SELECTION PAD, ON SELECTION POPUP, RESTORE, RETURN, SAVE, SAVE SCREEN, STORE

RENAME

Met het commando RENAME kunt u de naam van een bestand wijzigen.

Syntaxis

RENAME <oude bestandsnaam> TO <nieuwe bestandsnaam>

Standaardinstelling

Bij zowel de oude als de nieuwe bestandsnaam moet een extensie worden opgegeven. Als de bestanden niet op het standaardstation staan, moeten een pad en een stationsaanduiding worden opgegeven voor de bestandsnamen.

Gebruik

Met dit commando kunt u de namen van bestanden wijzigen zonder naar DOS te gaan. U kunt een geopend bestand geen andere naam geven en de nieuwe bestandsnaam mag geen naam zijn die al in dezelfde bestandsindex staat.

Als u een database-bestand met een bijbehorend bestand (bijvoorbeeld een memo- of een indexbestand) een andere naam geeft, moet u de bijbehorende bestanden dezelfde nieuwe naam geven. Het is aan te bevelen een database-bestand (en de bijbehorende bestanden) een andere naam te geven met MODIFY STRUCTURE en het op te slaan onder een nieuwe naam. Hiermee krijgt het bijbehorende mdx-bestand echter geen nieuwe naam.

Speciaal geval

De letters A tot en met J en M mogen niet worden gebruikt als naam voor een database-bestand omdat deze letters zijn gereserveerd voor aliasnamen. AA is echter wel een geldige naam voor een database-bestand.

Voorbeelden

U kunt het database-bestand School.dbf als volgt de nieuwe naam Rooster.dbf geven:

```
.RENAME School.dbf TO Rooster.dbf
```

U kunt het bestand Brieven.jim als volgt de naam Memos.oud geven, gesteld dat B niet het standaardstation is:

```
.RENAME B:Brieven.jim TO B:Memos.oud
```

Zie ook

CLOSE, COPY, COPY FILE, MODIFY STRUCTURE, USE

REPLACE

Met het commando REPLACE kunt u de inhoud van bepaalde velden in het actieve database-bestand wijzigen.

Syntaxis

REPLACE <veld> WITH <uitdr> [ADDITIVE] [,<veld> WITH <uitdr>
[ADDITIVE]] [<bereik>] [FOR <voorwaarde>]
[WHILE <voorwaarde>]

Standaardinstelling

Alleen het actieve record wordt gewijzigd, tenzij anders wordt bepaald door het bereik of door de FOR- of WHILE-clausule.

Gebruik

Met REPLACE vervangt u de gegevens in een bepaald veld. U kunt elk type veld kiezen, ook een memoveld. Het veld en de uitdrukking WITH moeten echter van hetzelfde gegevenstype zijn. In numerieke velden mag de uitdrukking WITH groter zijn dan de veldlengte. In dat geval wordt het getal in wetenschappelijke notatie weergegeven. Wanneer memovelden worden omgezet in tekenvelden, worden de gegevens door REPLACE afgekapt en aan de lengte van het veld aangepast.

Met de clausule ADDITIVE kunt u een memoveld samenstellen uit meerdere tekenreeksen. De clausule wordt genegeerd als het veld geen memoveld is. Wanneer u met REPLACE gegevens in een memoveld vervangt, moet de inhoud bestaan uit tekens.

De optie <bereik> wordt gebruikt om NEXT[n] op te geven, dat wil zeggen het aantal velden dat moet worden vervangen.

OPMERKING
Wanneer de inhoud van sleutelvelden in een geïndexeerd database-bestand wordt vervangen, wordt het actieve indexbestand bijgewerkt. Nadat het veld is gewijzigd, wordt het record verplaatst naar een nieuwe positie in het indexbestand, als er tenminste een index geactiveerd is. Als u een bereik FOR of WHILE opgeeft bij het vervangen van de inhoud van geïndexeerde velden, kan het zijn dat niet alle records volgens verwachting worden gewijzigd. Als u bijvoorbeeld het commando REPLACE ALL geeft, worden alleen het eerste record en de records na de nieuwe veldwaarde vervangen. De oorzaak daarvan is dat elk record opnieuw wordt geïndexeerd nadat het is vervangen en dat de recordwijzer naar de nieuwe positie van het record wordt verplaatst. Om alle records te vervangen moet u de feitelijke volgorde van het database-bestand herstellen met SET ORDER TO ↵

Recordwijzer

De recordwijzer wordt alleen verplaatst als u een bereik of de clausule FOR of WHILE opgeeft. Als de recordwijzer aan het einde van het bestand staat (EOF() is .T.), worden geen records vervangen, tenzij u een bereik of een voorwaarde FOR opneemt in het commando.

Voorbeelden

In de volgende voorbeelden wordt het database-bestand Magazijn.dbf gebruikt. In het eerste voorbeeld worden alleen waarden vervangen in de records die voldoen aan de gestelde voorwaarde:

```
. USE Magazijn
. REPLACE FOR Ondrdl_nr = "C-222-1000" Stukprijs WITH 1300
    2 records vervangen
```

Op de volgende wijze kunt u de kostprijs in alle records met tien procent verhogen:

```
. REPLACE ALL Stukprijs WITH Stukprijs * 1.1
    17 records vervangen
```

Zie ook

ORDER, REINDEX, SET INDEX, SET ORDER, STORE

REPLACE FROM ARRAY

Met het commando REPLACE FROM ARRAY worden velden in een database-bestand vervangen door gegevens uit een reeks.

Syntaxis

REPLACE FROM ARRAY <naam reeks> [<bereik>]
[FIELDS]<veldenlijst>][FOR <voorwaarde>]
[WHILE<voorwaarde>]

Standaardinstelling

Het standaardbereik is het actieve record in het actieve database-bestand.

Gebruik

De gegevens uit de eerste rij van een opgegeven reeks worden gekopieerd naar de velden van het actieve database-bestand. Het eerste element uit de reeks wordt naar het eerste veld gekopieerd, het tweede element naar het tweede veld enzovoort. Het proces stopt wanneer er geen velden meer zijn om te vervangen of wanneer er geen elementen meer in de eerste rij van de reeks staan.

Met de FIELDS-clausule kunt u de velden opgeven die moeten worden vervangen. De relatie tussen velden in het database-bestand en elementen uit de reeks wordt bepaald door de volgorde van de velden in de veldenlijst.

Wanneer u een bereik opgeeft of met behulp van FOR of WHILE meerdere records, worden net zo lang velden in de opgegeven records vervangen totdat er geen elementen meer in de reeks staan.

Het gegevenstype van de elementen uit de reeks moet hetzelfde zijn als van de velden die zullen worden vervangen. Wanneer het gegevenstype niet hetzelfde is, wordt de foutmelding **Gegevenstype onjuist** weergegeven. Wijzigingen in velden die al verwerkt zijn worden niet meer ongedaan gemaakt. Memovelden kunnen niet worden vervangen, omdat ze niet als elementen van een reeks kunnen worden opgeslagen.

Database-bestanden (of de veldenlijst) kunnen meer velden bevatten dan het aantal kolommen van de reeks. In de extra velden worden in dat geval geen waarden gewijzigd. Als de reeks meer kolommen bevat dan het aantal velden uit het database-bestand of de veldenlijst, worden de extra kolommen genegeerd.

REPLACE FROM ARRAY houdt rekening met de mogelijk actieve SET FIELDS-lijst en zal alleen waarden vervangen in de velden van het database-bestand die zijn opgegeven in deze lijst.

Voorbeeld

In het volgende voorbeeld wordt de procedure getoond waarmee gegevens uit een record naar een reeks worden gekopieerd, vervolgens in het geheugen worden bewerkt en ten slotte worden teruggekopieerd naar het record.

```
<EXAMPLE>USE Magaz ORDER Order_nr
DECLARE mmagaz[1,6]
morder = SPACE(6)
CLEAR
@ 10, 10 SAY "Order?" " GET morder ;
        PICTURE "99-999"
READ
IF SEEK(morder)
    COPY TO ARRAY mmagaz NEXT 1
    CLEAR
    @  8, 13 SAY "Ordernr: " + mmagaz[1,1]
    @  9, 14 SAY "Art.nr:"      GET mmagaz[1,2]
    @ 10, 12 SAY "Artikel:"     GET mmagaz[1,3]
    @ 11, 10 SAY "Omschr.:"     GET mmagaz[1,4]
    @ 12, 12 SAY "Kosten:"      GET mmagaz[1,5] PICTURE "99999.99"
    @ 13, 13 SAY "Aantal:"      GET mmagaz[1,6] PICTURE "999"
    READ
    mok = .T.
    @ 15, 10 SAY "Wijzigingen opslaan?"    GET mok PICTURE "J"
    READ
    IF mok
        REPLACE FROM ARRAY mstock
    ENDIF
ENDIF
```

Zie ook

APPEND FROM ARRAY, COPY TO ARRAY, DECLARE, PUBLIC

REPORT FORM

Met REPORT FORM kunt u informatie afdrukken uit de actieve database of visie op basis van een rapportmodel dat is gemaakt met CREATE/MODIFY REPORT. U kunt het rapport op het scherm weergeven, afdrukken of in een bestand opslaan.

Syntaxis

REPORT FORM <naam formulierbestand>/? [PLAIN] [HEADING <Tuitdr>]
[NOEJECT] [SUMMARY] [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]
[TO PRINTER/TO FILE <bestandsnaam>]

Standaardinstelling

De standaardextensie voor een formulierbestand is .frm.

Alle records in de database worden opgenomen in het bestand, tenzij anders wordt bepaald door het bereik, een FOR-clausule, een actieve filtervoorwaarde of een actief query-bestand.

Gebruik

Maak eerst een rapportmodel met CREATE/MODIFY REPORT. Als records in het rapportmodel in groepen worden ingedeeld, moet de actieve database of visie zijn geïndexeerd of gesorteerd op de velden op basis waarvan de groepen worden ingedeeld.

Als het rapportmodel in dBASE III PLUS-indeling is opgeslagen, wordt het bestand eerst naar dBASE IV-indeling geconverteerd. Vervolgens wordt het nieuwe bestand opgeslagen met de extensie .frm en wordt een bestand gegenereerd met de extensie .frg. Daarna wordt het bestand gecompileerd naar een objectbestand met de extensie .fro. Dan pas wordt het rapport uitgevoerd op basis van het .fro-bestand. De extensie van het oude dBASE III PLUS-bestand wordt gewijzigd in .fr3.

Als u na het commando REPORT FORM een bestandsnaam opgeeft met de extensie .frg of .fro, wordt het rapport gemaakt op basis van het .frg- of fro-bestand. Geeft u een andere extensie op of laat u deze helemaal achterwege, dan zoekt dBASE IV eerst naar een .frg- of .fro-bestand met de opgegeven naam en wordt dat bestand vervolgens uitgevoerd. Als geen .fro- of .frg-bestand met de opgegeven naam wordt gevonden, wordt verondersteld dat het opgegeven bestand een ontwerpbestand is, dat is gemaakt met het commando CREATE/MODIFY REPORT. Er wordt in dat geval een .frg-bestand gegenereerd uit het ontwerpbestand voordat het rapport wordt uitgevoerd.

Wanneer u wilt dat de instellingen uit een specifiek afdrukmodel worden gebruikt in plaats van de standaardinstellingen van de actieve omgeving, moet u de naam van het bestand met dat afdrukmodel toevoegen aan de _pform-geheugenvariabele van het systeem.

Wijzigingen die in een .prf-bestand zijn aangebracht, kunnen alleen vanuit het rapportscherm van het Control Center aan het rapport worden toegevoegd.

Wanneer SET ECHO op ON is ingesteld, wordt de broncode van het .frg-bestand afgedrukt in plaats van het rapport.

In een netwerk wordt het database-bestand automatisch door REPORT FORM vergrendeld voordat het wordt afgedrukt als SET LOCK op ON is ingesteld. Als een andere gebruiker de functie RLOCK() of FLOCK() heeft gebruikt voor het bestand, geeft REPORT FORM de foutmelding Bestand in gebruik door een andere gebruiker. U kunt het bestand eventueel kopiëren naar een tijdelijk bestand om toch een rapport te genereren. REPORT FORM vergrendelt het bestand niet wanneer SET LOCK op OFF staat.

Opties

Met de vraag-clausule, ?, kunt u een menu activeren waarin alle rapportmodellen van het actieve database-bestand staan.

Als u de optie PLAIN opgeeft, wordt de kop- en voettekst van het rapport alleen op de eerste pagina afgedrukt.

Met de optie HEADING, gevolgd door een tekenuitdrukking, kunt u een extra koptekst definiëren die wordt afgedrukt op de eerste regel van iedere pagina. Als de koptekst een tekenreeks is, moet deze tussen aanhalingstekens worden geplaatst. PLAIN en HEADING kunnen niet samen worden gebruikt. PLAIN heeft dan de hoogste prioriteit.

Als NOEJECT samen met TO PRINTER wordt opgegeven, wordt de paginadoorvoercode onderdrukt die gewoonlijk aan het begin wordt gegeven. Hierdoor wordt het rapport zonder voorafgaande blanco pagina afgedrukt.

Wanneer u Nieuwe regel bij puntkomma instelt op ON bij het maken van een rapport, worden puntkomma's (;) in memo- of tekenvelden geïnterpreteerd als een wagenterugloop en regeldoorvoer wanneer het rapport wordt uitgevoerd. De puntkomma wordt niet afgedrukt.

Met de optie TO FILE slaat u het rapport op als tekstbestand (.txt) op het standaardstation. Het rapport wordt ook op het scherm weergegeven, tenzij anders wordt bepaald.

Met SUMMARY voorkomt u de weergave van detailregels. Alleen totalen en subtotalen verschijnen dan in het rapport.

Zie ook

CREATE/MODIFY REPORT, INDEX, LABEL FORM, SET LOCK, SORT, _pform

Hoofdstuk 10 van *De menu's van dBASE IV*

RESET

Met het commando RESET kunt u de integriteitsvlag van een bestand verwijderen.

Syntaxis

RESET [IN <alias>]

Gebruik

Bestanden die zijn betrokken bij een transactie die in gang is gezet met het commando BEGIN TRANSACTION, worden gemarkeerd met een integriteitsvlag. Deze vlag wordt pas verwijderd als de transactie is voltooid, of als er met succes een ROLLBACK is uitgevoerd.

Als u de wijzigingen in een bestand niet ongedaan wilt maken met ROLLBACK, of als het commando ROLLBACK geen succes heeft, kunt u met het commando RESET IN <alias> de integriteitsvlag verwijderen.

<alias> is de aliasnaam van een database-bestand dat geopend is in een ander werkgebied. Met de clausule IN kunt u een database-bestand in een ander werkgebied bewerken zonder dit werkgebied eerst te kiezen als het actieve werkgebied.

Na een RESET-commando kunt u een andere transactie uitvoeren met het bestand.

Maak in een programma geen gebruik van RESET. Geef dit commando alleen indien nodig bij de commandostip om een ongewone situatie te corrigeren.

Speciaal geval

In een netwerk moet het database-bestand in exclusief gebruik zijn voordat u het RESET-commando geeft.

Opties

Als <alias> kan worden gebruikt:

- Een getal tussen 1 en 10
- De letters A tot en met J
- Een aliasnaam, standaard of opgegeven met de optie ALIAS van het commando USE.
- Een numerieke uitdrukking waarvan de uitkomst een getal tussen 1 en 10 is (tussen haakjes als het een eenvoudige variabele is)
- Een tekenuitdrukking waarvan de uitkomst een van de letters A tot en met J of een aliasnaam is

Zie ook

BEGIN TRANSACTION, END TRANSACTION, ISMARKED(), ROLLBACK, ROLLBACK()

RESTORE

Met RESTORE kunt u geheugenvariabelen en reeksen ophalen uit een geheugenbestand en deze activeren.

Syntaxis

RESTORE FROM <bestandsnaam> [ADDITIVE]

Standaardinstelling

De standaardextensie .mem is van toepasing, tenzij u een andere extensie opgeeft.

Gebruik

Wanneer u geheugenvariabelen terugzet met RESTORE, worden alle variabelen gewist die op dat moment actief zijn, tenzij u de optie ADDITIVE opgeeft. In een dBASE-programmabestand worden deze variabelen teruggezet als PRIVATE variabelen, tenzij u de variabelen eerst als PUBLIC definieert en vervolgens de optie ADDITIVE opgeeft. Geheugenvariabelen die vanaf de commandostip worden teruggezet, worden altijd PUBLIC.

U hebt de beschikking over maximaal 25.000 geheugenvariabelen. U kunt het maximum aantal variabelen instellen in het bestand Config.db. Het standaardaantal is 500. Het kan echter zijn dat het maximum aantal variabelen wordt beperkt door het beschikbare geheugen. Raadpleeg hoofdstuk 6, "dBASE IV aanpassen", voor meer informatie over de standaardaantallen geheugenvariabelen en hoe u deze kunt wijzigen.

Opties

Geef de optie ADDITIVE op wanneer u de actieve geheugenvariabelen wilt behouden. Op deze manier worden de geheugenvariabelen die worden teruggezet, toegevoegd aan de actieve variabelen. Geheugenvariabelen met dezelfde naam worden overschreven. Als in een geheugenbestand de geheugenvariabelen als PUBLIC moeten worden teruggezet, definieert u de variabelen eerst als PUBLIC voordat u het geheugenbestand terugzet, en geeft u vervolgens de optie ADDITIVE op.

Zie ook

DECLARE, PRIVATE, PUBLIC, RELEASE, SAVE, STORE

RESTORE MACROS

Met het commando RESTORE MACROS worden macro's die in een macrobibliotheek zijn opgeslagen, in het geheugen teruggezet.

Syntaxis

RESTORE MACROS FROM <macrobestand>

Gebruik

Een macrobestand (.key) wordt gemaakt wanneer u het commando SAVE MACROS geeft, en bevat alle macro's die op dat moment in het geheugen aanwezig zijn. Met RESTORE MACROS kunt u die macro's vanuit het bestand terugzetten in het geheugen, zodat ze tijdens de actieve sessie van dBASE IV beschikbaar zijn.

Als u de macro's opslaat in een bestand dat niet de extensie .key heeft, moet u die extensie later ook opgeven wanneer u de macro's terugzet met RESTORE MACROS.

Wanneer u macro's vanuit een bestand terugzet in het geheugen, komen deze in de plaats van macro's die aan dezelfde toetsen zijn toegewezen als de macro's uit het macrobestand.

Voorbeeld

Met het volgende commando worden alle macro's in het opgegeven bestand teruggezet in het geheugen:

```
. RESTORE MACROS FROM Macdemo
```

Met dit commando worden alle macro's die zijn opgeslagen in het bestand Macdemo.key, teruggezet in het geheugen.

Zie ook

PLAY MACRO, SAVE MACROS

RESTORE SCREENS

Met het commando RESTORE SCREENS wordt een schermafbeelding vanuit de schermnaam-geheugenvariabele teruggezet.

Syntaxis

RESTORE SCREEN FROM<schermnaam>

Gebruik

De schermafbeelding moet in het geheugen zijn opgeslagen met het commando SAVE SCREEN TO<schermnaam>. Schermafbeeldingen kunnen vanuit een één maal opgeslagen schermafbeeldingen vaker worden teruggezet.

Opgeslagen schermen worden met RELEASE SCREENS of CLEAR ALL uit het geheugen gewist.

Zie ook

ACTIVATE SCREEN, CLEAR ALL, CLEAR SCREENS, RELEASE SCREEN, SAVE SCREEN

RESTORE WINDOW

Met het commando RESTORE WINDOW zet u vensterdefinities vanuit een bestand terug in het geheugen.

Syntaxis

RESTORE WINDOW <lijst vensternamen>/ALL FROM <bestandsnaam>

Gebruik

Met dit commando kunt u vensterdefinities terugzetten die in een bestand zijn opgeslagen met het commando SAVE WINDOW. De standaardextensie voor een vensterdefinitiebestand is .win.

U kunt vensters selectief terugzetten. U hoeft dus niet alle vensternamen in een bepaald .win-bestand terug te zetten. Ook kunt u vensters in een willekeurige volgorde terugzetten, ongeacht de volgorde waarin deze zijn opgeslagen met het commando SAVE WINDOW. Als u de optie ALL opgeeft, worden alle vensterdefinities in het .win-bestand teruggezet in het geheugen.

Als u een andere extensie dan .win hebt opgegeven toen u de vensterdefinities opsloeg, moet u die extensie ook opgeven bij het commando RESTORE WINDOW.

Wanneer u een venster terugzet dat dezelfde naam heeft als een venster dat op dat moment in het geheugen aanwezig is, wordt dit laatste venster overschreven.

Zie ook

ACTIVATE WINDOW, DEFINE WINDOW, SAVE WINDOW

RESUME

Het commando RESUME is het complement van het commando SUSPEND. Een programma dat is onderbroken met SUSPEND, kan worden hervat met RESUME.

Syntaxis

RESUME

Gebruik

Wanneer u het commando RESUME geeft, wordt het eerder onderbroken commandobestand verder uitgevoerd. De eerste regel die wordt uitgevoerd, is de regel die volgt op de regel die werd uitgevoerd toen het programma werd onderbroken. Als u het commando ROLLBACK hebt gebruikt terwijl een transactie met SUSPEND was onderbroken, wordt het programma hervat op de regel die volgt op het commando END TRANSACTION in het programma.

RESUME werkt ook nadat een DEBUG-commando is onderbroken.

Tip

Het is raadzaam het commando CLEAR te geven voordat u RESUME gebruikt. Hierdoor kunnen de commando's die u hebt gegeven terwijl het programma was onderbroken, geen invloed hebben op de uitvoer naar het beeldscherm.

Zie ook

BEGIN TRANSACTION, CANCEL, CLEAR, DEBUG, RETURN, ROLLBACK, SUSPEND

RETRY

Wanneer u het commando RETRY geeft, wordt een commando waarin zich een fout voordeed, opnieuw uitgevoerd.

Syntaxis

RETRY

Gebruik

RETRY dient hoofdzakelijk om fouten te herstellen, omdat u op deze wijze een commando kunt herhalen totdat het met succes is uitgevoerd. Houd er echter rekening mee dat een IF- of DO WHILE-voorwaarde niet opnieuw wordt getoetst. Dit in tegenstelling tot RETURN waarmee u verplaatst tot de eerstvolgende regel na de fout.

RETRY roept het commando aan waarmee de laatste procedure is gestart. Met behulp van de procedure die u start met ON ERROR DO <procedure> wordt de fout geïsoleerd en gecorrigeerd. Vervolgens kunt u met RETRY terugkeren naar de commandoregel die de fout heeft veroorzaakt, waarna u het commando opnieuw uitvoert. Met RETRY wordt ook de waarde gewist die de functie ERROR() heeft opgeleverd.

Voorbeelden

De volgende twee commandobestanden laten zien hoe u de foutmelding **Bestand is al geopend** (foutnummer 3) kunt herstellen:

```
* Hoofd.prg
ON ERROR DO Herstel WITH ERROR()
USE Mijnbest
RETURN
* EOF:Hoofd.prg

* Herstel.prg
PARAMETERS Fout
IF Fout=3
    CLOSE DATABASES
ENDIF
RETRY
RETURN
* EOF:Herstel.prg
```

Zie ook

ERROR(), ON ERROR, RETURN

RETURN

In programma's dient het commando RETURN om terug te gaan naar het aanroepende programma, het Control Center of de commandostip. Wanneer de besturing wordt teruggegeven aan een programma, wordt het commando uitgevoerd dat volgt op het aanroepende commando. RETURN kan ook worden gebruikt aan het einde van een door de gebruiker gedefinieerde functie om het resultaat van die functie terug te geven.

Syntaxis

RETURN [<uitdrukking> / TO MASTER / <naam procedure>]

Gebruik

RETURN wordt gebruikt om een procedurebestand te sluiten en om de aanroepende procedure of functie te hervatten, of om naar de commandostip terug te gaan. Als u de optie TO MASTER opneemt in een programma, wordt het aanroepende programma van het hoogste niveau hervat. Als u de naam van een procedure opgeeft in plaats van de optie TO MASTER, wordt een procedure hervat die op dat moment actief is. De optie MASTER heeft een hogere prioriteit dan een procedure met de naam "master".

Het is noodzakelijk dat u het commando RETURN opneemt in een procedure of functie, omdat dBASE IV anders niet weet welke waarde moet worden opgeleverd.

Met RETURN worden alle PRIVATE geheugenvariabelen opgeheven die in de betrokken procedure zijn gedefinieerd. PUBLIC variabelen worden niet gewijzigd. Wanneer de functie ERROR() in een waarde heeft geresulteerd, wordt die waarde ook gewist. Oude waarden van PRIVATE systeemgeheugenvariabelen worden hersteld.

RETURN wordt gebruikt aan het einde van een door de gebruiker gedefinieerde functie om het resultaat van de functie weer te geven. Gebruik hiervoor de opdracht <uitdrukking>. Er moet een <uitdrukking> worden opgegeven bij RETURN wanneer het commando wordt gecompileerd als een door de gebruiker gedefinieerde functie.

Met een RETURN uit een ON ERROR-routine gaat u naar de regel die volgt op de fout.

Zie ook

COMPILE, DEBUG, DO, ERROR(), FUNCTION, PARAMETER, PRIVATE, PROCEDURE, PUBLIC, RESUME, SUSPEND

ROLLBACK

Het commando ROLLBACK brengt de database- en indexbestanden terug in de staat van voor de actieve transactie en beëindigt vervolgens de transactie. Alle bestanden die tijdens de transactie zijn gemaakt, worden gesloten en verwijderd.

Syntaxis

ROLLBACK [<naam database-bestand>]

Gebruik

Met dit commando maakt u wijzigingen ongedaan die in records van een database-bestand zijn aangebracht nadat het commando BEGIN TRANSACTION is gegeven.

Als u het commando ROLLBACK geeft, wordt de transaction-log vergeleken met de inhoud van de actieve database om de actieve transactie ongedaan te maken en het database-bestand in de oorspronkelijke staat te herstellen.

Tijdens een transactie kan het commando ROLLBACK niet worden gebruikt met de toevoeging <bestandsnaam>. ROLLBACK dient alleen te worden gebruikt om systeemfouten te herstellen. U moet altijd eerst het commando zonder bestandsnaam geven. Als het onmogelijk blijkt, alle database-bestanden die tijdens een transactie zijn gewijzigd in een keer te verwerken, kunt u met ROLLBACK<bestandsnaam> proberen de database-bestanden per stuk te herstellen.

Nadat een transactie met ROLLBACK ongedaan is gemaakt, wordt de transaction-log verwijderd en berust de besturing bij het commando dat na END TRANSACTION staat.

In de transaction-log staat de inhoud van ieder gewijzigd record voordat en nadat het is gewijzigd. Op basis van deze transaction-log worden transacties ongedaan gemaakt. Als ROLLBACK geen effect heeft, kunnen daar twee oorzaken voor zijn:

- De inhoud van een record voor de transactie is strijdig met de inhoud van dat record na de transactie.
- De transaction-log kan niet worden gelezen.

Voorbeeld

In dit deel van het programma wordt een automatische ROLLBACK geactiveerd met het commando ON ERROR, voordat een transactie wordt gestart waarmee de salarisrecords worden aangepast:

```
ON ERROR ROLLBACK
BEGIN TRANSACTION
    REPLACE ALL Salaris WITH Salaris * 1.1
END TRANSACTION
ON ERROR
```

Zie ook

BEGIN/END TRANSACTION, COMPLETED(), ISMARKED(), RESET, ROLLBACK()

RUN

Met RUN kunt u vanuit dBASE IV een DOS-commando geven of een ander programma aanroepen dat in DOS wordt uitgevoerd.

Syntaxis

RUN/! <DOS-commando>

Gebruik

Met RUN, of de alternatieve syntaxis !, kunt u een programma of een DOS-commando uitvoeren. Wanneer de uitvoering hiervan is beëindigd, keert u terug naar dBASE IV.

Het commando RUN is alleen bruikbaar als er genoeg geheugen over is voor het DOS-bestand Command.com en voor het commando of programma zelf. Als er in het werkgeheugen niet genoeg ruimte over is, verschijnt de foutmelding **Onvoldoende geheugen**.

Wanneer RUN in een venster wordt uitgevoerd, zal de uitvoer van het programma of het commando op het scherm blijven staan tot u op een willekeurige toets drukt. Dit om u de gelegenheid te geven de uitvoer te bekijken voordat het scherm wordt gewist.

Het DOS-bestand Command.com moet zich bevinden in de actieve bestandsindex of in een bestandsindex op het pad dat wordt opgegeven met de DOS-parameter COMSPEC.

Met het DOS-commando SET COMSPEC kunt u het besturingssysteem opgeven waar Command.com zich bevindt (bijvoorbeeld SET COMSPEC=C:\COMMAND.COM). Raadpleeg het boek bij DOS voor meer informatie over SET en COMSPEC.

WAARSCHUWING
Sommige DOS-commando's, zoals ASSIGN en PRINT, blijven in het werkgeheugen aanwezig nadat zij zijn uitgevoerd. Voer deze commando's daarom uit voordat u dBASE IV laadt. ASSIGN vergrendelt uw computer zonder een foutmelding. Als u de foutmelding Onvoldoende geheugen krijgt, kunt u proberen TSR-programma's te verwijderen of de computer opnieuw opstarten en het opnieuw proberen.

Voorbeeld

Op de volgende wijze kunt u de systeemdatum wijzigen met behulp van een geheugenvariabele:

```
. Vandaag = "29-02-88"
29-02-88
. RUN DATE &Vandaag.
. ? DATE()
29-02-88
```

Zie ook

CALL, CALL(), GETENV(), LOAD, RUN()

SAVE

Met het commando SAVE kunt u de actieve verzameling geheugenvariabelen en reeksen of een deel daarvan opslaan in een bestand.

Syntaxis

SAVE TO <bestandsnaam> [ALL LIKE/EXCEPT <structuur>]

Standaardinstelling

Tenzij anders opgegeven wordt voor ieder nieuw bestand de extensie .mem gebruikt. Als u de optie ALL LIKE/EXCEPT niet opgeeft, worden alle actieve geheugenvariabelen opgeslagen.

Gebruik

De alternatieve syntaxis voor dit commando is:

SAVE ALL LIKE <structuur> TO <Tuitdr>

Met SAVE TO kunt u geheugenvariabelen opslaan die u in latere dBASE IV-sessies wilt gebruiken.

Voorbeelden

Op de volgende wijze kunt u alle geheugenvariabelen opslaan in het bestand Mijnbest.mem op station B:

```
. SAVE TO B:Mijnbest
```

Op de volgende manier slaat u alle geheugenvariabelen waarvan de naam begint met de letter *d*, op in het bestand Mijnbest.mem op station B:

```
. SAVE ALL LIKE d* TO B:Mijnbest
```

Met het volgende commando slaat u alle geheugenvariabelen op in het bestand Mijnbest.oud op het standaardstation, behalve de variabelen waarvan de letter *x* het vierde teken in de naam is:

```
. SAVE ALL EXCEPT ???x* TO Mijnbest.oud
```

Zie ook

DECLARE, PARAMETERS, PRIVATE, PUBLIC, RESTORE, SET SAFETY, STORE

SAVE MACROS

Met SAVE MACROS kunt u de macro's die op dat moment zijn gedefinieerd, vanuit het geheugen opslaan in een bestand. U kunt het macrobestand in het geheugen terugzetten op het moment dat u de bewuste verzameling macrodefinities weer wilt gebruiken.

Syntaxis

SAVE MACROS TO <macrobestand>

Gebruik

Met dit commando maakt u een bestand met de standaardextensie .key. U kunt de macrotoetsen **Alt-F1** tot en met **Alt-F9** definiëren en ook **Alt-F10**, gevolgd door een van de letters A tot en met Z. Er zijn dus 35 unieke macrotoetsen.

Een macrobestand bevat minimaal 1 macrodefinitie en maximaal 35 macrodefinities.

Als SET SAFETY op ON is ingesteld en er reeds een macrobestand met de opgegeven naam bestaat, verschijnt er een melding waarin u wordt gevraagd of u het betrokken bestand inderdaad wilt overschrijven.

U kunt ook een andere extensie dan .key opgeven, maar dan moet u die extensie ook opgeven als u het bestand in het geheugen terugzet.

Voorbeeld

```
. AVE MACROS TO Macdemo
```

In het voorbeeld wordt een macrobestand gemaakt met de naam Macdemo.key, dat de actieve macrodefinities bevat.

Zie ook

KEYBOARD, PLAY MACRO, RESTORE MACROS

SAVE SCREEN

Met het commando SAVE SCREEN kunt u het actieve scherm opslaan onder de naam van een geheugenvariabele

Syntaxis

SAVE SCREEN TO <schermnaam>

Gebruik

U kunt SAVE SCREEN in een programma gebruiken om een beeld van het scherm in het geheugen op te slaan. Het beeld kan later opnieuw worden weergegeven met RESTORE SCREEN. SAVE SCREEN is niet verwant met het commando ACTIVE SCREEN.

Indien u een schermnaam gebruikt die al aan een eerder opgeslagen scherm is gegeven, wordt het oorspronkelijke scherm overschreven.

Opgeslagen schermen worden met het commando RELEASE SCREENS en CLEAR SCREENS uit het geheugen verwijderd.

Het commando SAVE SCREEN heeft geen invloed op de dBASE IV-bestanden voor schermindeling (.scr, .fmt, .fmo).

VOORBEELD

In het volgende voorbeeld wordt getoond hoe het actieve scherm wordt opgeslagen en een melding wordt weergegeven. Nadat de gebruiker op een toets heeft gedrukt, wordt het oorspronkelijke beeld hersteld en het programma voortgezet.

```
postbest = USER() + ".box"
IF FILE(postbest)
    SAVE SCREEN TO postopsl
    @ 10, 10 TO 14, 69 DOUBLE
    @ 11, 11 CLEAR TO 13, 68
    @ 12, 12 SAY "U hebt post. Druk op een toets."
    tijdtoets = INKEY(0)
    RESTORE SCREEN FROM postopsl
    RELEASE SCREEN postopsl
ENDIF
```

Zie ook

CLEAR SCREENS, RELEASE SCREENS, RESTORE SCREEN

SAVE WINDOW

Met het commando SAVE WINDOW kunt u een vensterdefinitie opslaan in een bestand.

Syntaxis

SAVE WINDOW <vensternamen>/ALL TO <bestandsnaam>

Gebruik

Met dit commando kunt u vensters opslaan die u hebt gedefinieerd, zodat u ze later weer kunt gebruiken zonder ze opnieuw te hoeven definiëren. U kunt een set vensters definiëren, de vensters gebruiken en ze vervolgens met het commando CLEAR verwijderen totdat u ze weer wilt gebruiken.

Als u het commando SAVE WINDOW geeft, wordt het opgegeven bestand gemaakt en worden alle vensters in het geheugen in dit bestand opgeslagen, of alleen de vensters die worden opgegeven in <vensternamen>. Alleen vensters die zijn gedefinieerd en op dat moment in het geheugen staan, kunnen worden opgeslagen.

Als er al een bestand met dezelfde naam bestaat, wordt dat bestand overschreven. Als u SET SAFETY op ON hebt ingesteld, wordt u gewaarschuwd voordat het bestand wordt overschreven.

De standaardextensie voor het uitvoerbestand is .win. Als u een andere extensie gebruikt voor het vensterdefinitiebestand, moet u die extensie ook opgeven bij het commando RESTORE WINDOW.

Als u de optie ALL gebruikt, worden alle in het geheugen aanwezige vensters in het bestand opgeslagen.

Zie ook

ACTIVATE WINDOW, CLEAR, DEFINE WINDOW, RESTORE WINDOW

SCAN

Het commando SCAN is een programmeerconstructie met een lusbewerking en een impliciete overslaan-opdracht, geschikt om incrementeel verwerkingscommando's op database-records toe te passen.

Syntaxis

SCAN [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]
[<commando's>]
[LOOP]
[EXIT]
ENDSCAN

Gebruik

Als u het commando SCAN geeft, wordt het actieve database-bestand doorlopen en worden de opgegeven commando's uitgevoerd voor records die binnen het <bereik> vallen en die voldoen aan de FOR- en WHILE-voorwaarden. Alle commando's die tussen SCAN en ENDSCAN staan, worden uitgevoerd voor de gekozen records.

De standaardinstelling van de optie <bereik> is het gehele bestand, te beginnen met het eerste record. Als u de optie NEXT of RECORD opgeeft als <bereik>, begint het commando bij het actieve record.

Met FOR <voorwaarde> kunt u bepalen voor welke records in het <bereik> de commando's moeten worden uitgevoerd.

Met WHILE <voorwaarde> kunt u de records nog verder beperken. Records mogen alleen worden bewerkt zolang WHILE <voorwaarde> waar (.T.) blijft.

Met LOOP wordt de besturing teruggegeven aan het begin van het SCAN-proces.

Met EXIT wordt een SCAN-proces beëindigd. De besturing wordt teruggegeven aan het eerstvolgende commando na ENDSCAN.

Recordwijzer

Wanneer het commando SCAN wordt gegeven, heeft dat bepaalde gevolgen voor de recordwijzer. Een SCAN begint aan het begin van een bestand (of met het eerste record in de index), tenzij u iets anders opgeeft met het <bereik> of met de FOR- en WHILE-voorwaarden. Een SCAN gaat door tot aan het einde van een bestand (of de laatste record in de index), tenzij het <bereik>, de FOR- of WHILE-voorwaarde, of de optie EXIT het proces eerder beëindigen. De recordwijzer blijft staan op het record waar het proces werd beëindigd. Zodra een SCAN aan het eind van een bestand komt (of aan de FOR- of WHILE-voorwaarde niet meer wordt voldaan), wordt het eerstvolgende commando na ENDSCAN uitgevoerd.

Aangezien SCAN zelf verplaatsingscommando's voor de recordwijzer geeft, is het niet nodig een SKIP-commando toe te voegen.

Voorbeelden

In dit eenvoudige voorbeeld van SCAN wordt een procedure uitgevoerd waarin openstaande orders worden gemaakt voor ieder record in het gegevensbestand Transact dat niet is gefactureerd:

```
USE Transact
SCAN FOR .NOT. Factuur
    DO Openordr WITH Bestel_nr, Klant_nr, Dat_trans
ENDSCAN
```

U kunt ook een DO WHILE gebruiken om hetzelfde resultaat te bereiken als in het bovenstaande voorbeeld. Hiervoor gebruikt u de volgende code:

```
USE Transact
LOCATE FOR .NOT. Factuur
DO WHILE .NOT. EOF()
    DO Openordr WITH Bestel_nr, Klant_nr, Dat_trans
    CONTINUE
ENDDO
```

Zie ook

CONTINUE, LOCATE

SEEK

Met het commando SEEK kunt u het eerste record van een geïndexeerd database-bestand opzoeken waarvan de sleutel overeenkomt met de opgegeven uitdrukking. Het commando SEEK stelt u in staat een database snel te doorzoeken.

Syntaxis

SEEK <uitdr>

Gebruik

SEEK wordt gewoonlijk gebruikt met geheugenvariabelen. De uitdrukking die bij SEEK wordt opgegeven, kan elke geldige uitdrukking zijn die van hetzelfde gegevenstype is als de indexsleutel.

Zoeken naar subtekenreeksen of delen van de sleutel werkt alleen als SET EXACT op OFF is ingesteld en de zoekuitdrukking overeenkomt met de indexsleutel, te beginnen met het meeste linkse teken.

Als de indexsleutel bijvoorbeeld <CUR>Smith, John<SIEF> is en SET EXACT op OFF is ingesteld, wordt de indexsleutel gevonden als u <CUR>Smi<SIEF> opgeeft. Als SET EXACT echter op ON is ingesteld, moet de zoekuitdrukking overeenkomen met de hele indexsleutel. In dit voorbeeld is dat <CUR>Smith, John<SIEF>.

Bij het commando SEEK wordt ook rekening gehouden met de beperkingen die voor de records zijn ingesteld met de commando's SET DELETED en SET FILTER.

Om records te vinden waarvan de sleutel exact overeenkomt met de zoekuitdrukking, moet u SEEK gebruiken in combinatie met SET NEAR ingesteld op ON. Gebruik de functie FOUND() om te controleren of de overeenkomst exact was of niet.

Als u het commando SEEK gebruikt terwijl SET NEAR op ON is ingesteld, levert FOUND() waar (.T.) op bij volledige overeenkomst tussen zoekuitdrukking en sleutel, en onwaar (.F.) als bij benadering overeenkomst bestaat. In het laatste geval wordt de recordwijzer naar het record verplaatst waarvan de sleutel direct naast de gezochte waarde ligt.

Als u het commando SEEK gebruikt terwijl SET NEAR op OFF is ingesteld, levert FOUND onwaar (.F.) op als geen enkele sleutel exact overeenkomt met de zoeksleutel. De recordwijzer wordt dan naar het einde van het bestand (EOF) verplaatst.

Voorbeeld

In het volgende voorbeeld wordt het database-bestand Transact gebruikt:

Om een <uitdrukking> van het type teken te zoeken:

```
. USE Transact ORDER Bestel_nr
. SEEK "87-115"
. ? Klant_nr, Factuur, Totaal_rek
C00001 .F.          165,00
```

Zie ook

CONTINUE, EOF(), FIND, FOUND(), INDEX, KEY(), LOCATE, MDX(), NDX(), SEEK(), SET DELETED, SET EXACT, SET FILTER, SET INDEX, SET NEAR, SET ORDER, TAG(), USE

SELECT

Met SELECT kunt u een werkgebied kiezen waarin een database-bestand moet worden geopend, of een werkgebied waarin al een database-bestand is geopend.

Syntaxis

SELECT <werkgebied/alias>

Gebruik

Wanneer u dBASE IV start, is werkgebied 1 het actieve werkgebied. Geldige werkgebieden zijn 1 tot en met 10 of A tot en met J. U kunt een database-bestand openen in ieder werkgebied dat u kiest. Werkgebied 10 is gereserveerd voor een geopende catalogus. U kunt geen database-bestand openen in dit werkgebied als een catalogus is geopend. Als u een database-bestand hebt geopend in werkgebied 10 en u daarna een catalogus opent, wordt het database-bestand automatisch gesloten. Als u een bestand opent in een werkgebied waarin al een geopend database-bestand staat, wordt het eerste bestand altijd gesloten.

Elk werkgebied wordt ook gebruikt om andere bestanden te openen die horen bij het database-bestand, zoals index-, query- en indelingsbestanden. Deze bijbehorende bestanden kunnen echter niet in meerdere werkgebieden tegelijk worden geopend.

Met SELECT kunt u van werkgebied veranderen zodat het geopende database-bestand in het nieuwe werkgebied het actieve database-bestand wordt. U kunt van werkgebied veranderen door de alias op te geven van een geopend database-bestand in een ander werkgebied, of het nummer of de letter van het werkgebied. Als u geen alias opgeeft als u het bestand opent, wordt de bestandsnaam (zonder extensie) de standaard aliasnaam. De letters A tot en met J, afhankelijk van het actieve werkgebied, worden gebruikt als de standaard aliasnaam wanneer de bestandsnaam geen geldige alias is.

In dBASE IV is het nog steeds mogelijk de macrofunctie (&) te gebruiken om met een variabele een werkgebied of de bijbehorende alias te kiezen, net als in dBASE III PLUS. dBASE IV kent echter een eenvoudige en veel snellere methode. Om een werkgebied te kiezen met een variabele moet u de variabele tussen haakjes zetten, waardoor de variabele een uitdrukking wordt. Bijvoorbeeld, als N=1, gebruikt u SELECT (N). Als u de variabele gebruikt in een uitdrukking kunt u de haakjes weglaten, bijvoorbeeld SELECT N+0.

Normaal gesproken gebruiken commando's de velden in het actieve database-bestand. U kunt echter ook velden opgeven in een ander werkgebied met behulp van een alias. dBASE IV kent verschillende commando's en functies waarmee andere werkgebieden kunnen worden opgegeven.

Als u het commando SET FIELDS hebt gegeven en SET FIELDS op ON is ingesteld, kunt u ook gegevens weergeven en wijzigen van velden die zijn opgegeven in de veldenlijst van andere werkgebieden.

Recordwijzer

Elk werkgebied heeft een aparte recordwijzer. Wanneer u van werkgebied verandert, heeft dat geen gevolgen voor de positie van de recordwijzers, tenzij tussen beide een relatie is gedefinieerd.

Voorbeelden

In dit voorbeeld wordt het database-bestand Klanten geopend in werkgebied 1 en wordt het database-bestand Transact geopend in werkgebied 3:

```
. SELECT 1
. USE Klanten ALIAS Namen
. SELECT 3
. USE Transact
```

U geeft de alias van het bestand of het nummer of de letter van het werkgebied op bij SELECT als u van werkgebied wilt veranderen:

```
. SELECT Namen
```

In deze voorbeelden hebt u de werkgebieden gekozen door het nummer van een werkgebied te gebruiken of de aliasnaam van het geopende database-bestand in een werkgebied. U kunt een werkgebied ook kiezen met een letter. In plaats van werkgebied 1 kiest u dan werkgebied A.(SELECT A).

Zie ook

ALIAS(), CLOSE, CONTINUE, LOCATE, SELECT(), SET CATALOG, SET FIELDS, SET VIEW, USE

SHOW MENU

Met het commando SHOW MENU geeft u een balkmenu weer zonder het te activeren.

Syntaxis

SHOW MENU <menunaam> [PAD <stripnaam>]

Gebruik

Met dit commando geeft u het in het commando opgegeven menu weer, over eventuele andere informatie op het scherm.

Met de optie PAD kunt u een bepaalde strip markeren van het menu dat wordt weergegeven.

Als u een balkmenu weergeeft met het commando SHOW MENU, kunt u de cursor niet verplaatsen en ook geen keuze maken. Met dit commando kunt u bij het ontwikkelen van een programma controleren of het menu wordt weergegeven zoals u het hebt ontworpen.

Zie ook

ACTIVATE MENU, CLEAR, CLEAR MENU, DEFINE MENU, DEFINE PAD

SHOW POPUP

Met het commando SHOW POPUP geeft u een pop-up menu weer zonder het te activeren.

Syntaxis

SHOW POPUP <naam pop-up>

Gebruik

Met dit commando kunt u een pop-up menu weergeven zonder het te activeren. Met deze functie kunt u de weergave van een pop-up controleren bij het ontwikkelen van programma's.

Omdat het pop-up menu niet actief is, wordt de cursor niet verplaatst en worden meldingen die bij het pop-up menu horen niet weergegeven.

Voorbeeld

Als u het pop-up menu Visie_pop wilt weergeven:

```
. SHOW POPUP Visie_pop
```

Het menu Visie_pop wordt weergegeven. Met het commando CLEAR kunt u het pop-up menu van het scherm verwijderen.

Zie ook

ACTIVATE POPUP, CLEAR, DEFINE BAR, DEFINE POPUP, POPUP()

SKIP

Met het commando SKIP kunt u de recordwijzer vooruit of achteruit verplaatsen in het database-bestand.

Syntaxis

SKIP [<Nuitdr>] [IN <alias>]

Gebruik

Als het database-bestand is geïndexeerd, volgt SKIP de indexvolgorde van de records. Als het bestand niet is geïndexeerd, wordt de recordwijzer sequentieel verplaatst in het bestand. Als het commando SET FILTER is gegeven, wordt rekening gehouden met de opgegeven voorwaarden.

De recordwijzer gaat vooruit in het database-bestand vanaf het actieve record tot aan het einde van het bestand, tenzij de uitkomst van de numerieke uitdrukking negatief is. Wanneer de uitdrukking negatief is, gaat de recordwijzer achteruit in het bestand vanaf het actieve record tot aan het begin van het bestand. Als er geen uitdrukking wordt opgegeven, gaat de recordwijzer één record vooruit.

Als u de clausule IN <alias> opgeeft, verplaatst SKIP de recordwijzer van het werkgebied met die alias. Met de clausule IN kunt u de recordwijzer uit een ander werkgebied verplaatsen zonder dat u dat werkgebied eerst hoeft te kiezen als het actieve werkgebied. De recordwijzer van het oorspronkelijke werkgebied wordt niet verplaatst. Wanneer SET TALK op ON is ingesteld, wordt de naam van het database-bestand weergegeven voor het recordnummer.

Recordwijzer

Als de recordwijzer op het laatste record in het bestand staat wanneer u het commando SKIP geeft, is de waarde van RECNO() het aantal records in het bestand plus 1 en is EOF() waar (.T.). Als u vervolgens SKIP invoert, levert dat een foutmelding op

Als u het commando SKIP -1 geeft terwijl de recordwijzer op het eerste record staat, blijft RECNO() 1. BOF() is echter waar (.T.), omdat de recordwijzer aan het begin van het bestand staat.

Voorbeelden

In deze voorbeelden worden de database-bestanden Afnemers en Transact gebruikt.

U kunt als volgt naar het volgende record in een database-bestand gaan:

```
. USE Afnemers
. SKIP
AFNEMERS: Recordnr.        2
```

U kunt op de volgende manier vijf records vooruit gaan met behulp van een numerieke geheugenvariabele:

```
. Moversl = 5
                    5
. SKIP Moversl
AFNEMERS: Recordnr.  7
```

Als Afnemers het actieve bestand is, kunt u op de volgende manier de recordwijzer te verplaatsen in het database-bestand Transact:

```
. USE Transact IN 2
. SKIP 4 IN Transact
TRANSACT: Recordnr.  5
```

Zie ook

BOF(), EOF(), GO/GOTO, RECNO(), SET SKIP

SORT

Met het commando SORT wordt een nieuw database-bestand aangemaakt waarin de records van het actieve database-bestand worden opgenomen in alfabetische, chronologische of numerieke volgorde, gesorteerd op basis van de opgegeven sleutelvelden.

Syntaxis

SORT TO <bestandsnaam> ON <veld1> [/A] [/C] [/D]
[,<veld2> [/A] [/C] [/D] ...] [ASCENDING]/[DESCENDING]
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstellingen

Met SORT sorteert u in oplopende volgorde (/A), tenzij u /D of DESCENDING (afdalend) opgeeft. Oplopende volgorde is hetzelfde als alfabetische volgorde (raadpleeg bijlage G).

Als er een catalogus wordt gebruikt, wordt deze bijgewerkt.

Gebruik

U kunt niet sorteren op logische velden of memovelden.

U kunt maximaal op tien velden sorteren.

Een bestand kan niet naar zichzelf of naar een ander geopend bestand worden gesorteerd.

Opties

Met de optie /C kunt u opgeven dat bij het sorteren geen verschil wordt gemaakt tussen hoofdletters en kleine letters. Met /C wordt dus een bestand gemaakt dat niet noodzakelijk in ASCII-volgorde staat.

De opties /A en /D hebben slechts invloed op één veld. U kunt /C combineren met /A of /D. Als u twee opties gebruikt, hoeft u maar één schuine streep te gebruiken (bijvoorbeeld /DC).

ASCENDING en DESCENDING geldt voor alle velden die niet de optie /A of /D hebben. De opties /A en /D hebben een hogere prioriteit dan de opties ASCENDING en DESCENDING, maar alleen voor de veldnaam die voor de optie /A of /D staat.

Tips

U kunt de records in een database-bestand opnieuw ordenen met de commando's SORT en INDEX. Met SORT wordt een nieuw database-bestand gemaakt dat afzonderlijk wordt opgeslagen. Met INDEX wordt een indexbestand (.ndx) gemaakt of een label in een multi-indexbestand (.mdx) waarin wijzers staan naar elk record in het originele bestand, maar de fysieke volgorde van het originele database-bestand wordt niet gewijzigd. U kunt het commando SORT beschouwen als een INDEX-commando dat wordt gevolgd door COPY.

De volgende commando's:

```
. INDEX ON Achternaam TO Anaam
. COPY TO Naambest
```

lijken veel op

```
.SORT ON Achternaam TO Naambest
```

Zowel met SORT als met INDEX worden de records in ASCII-volgorde gesorteerd.

Bij het commando SORT kunt u alleen een eenvoudig veld of een eenvoudige veldenlijst opgeven, terwijl bij het commando INDEX elke geldige uitdrukking kan worden opgegeven (ook eenvoudige velden). Als u wilt sorteren op een uitdrukking (wat niet kan met SORT), geeft u het commando INDEX met de gewenste uitdrukking en vervolgens het commando COPY om een nieuw gesorteerd bestand te maken.

Afhankelijk van uw toepassing kan het soms wenselijk zijn om de fysieke volgorde van de gegevens te wijzigen met SORT en soms om een index te maken zonder het database-bestand te wijzigen. Er zijn nog twee factoren die uw keuze kunnen bepalen:

- De commando's FIND en SEEK kunnen alleen worden gebruikt voor geïndexeerde bestanden. Als u in een database-bestand wilt zoeken naar records of velden die overeenkomen met een opgegeven waarde, moet u INDEX gebruiken.
- Het commando INDEX kent twee opties, te weten UNIQUE en FOR. Door toevoeging van de optie UNIQUE wordt alleen het eerst voorkomende record in de index opgenomen als er meerdere records met dezelfde sleutelwaarde voorkomen. Door de optie FOR te gebruiken, kunt u in de index een subset opnemen van de records die aan een voorwaarde voldoen. De sorteervolgorde van de records in een met INDEX gemaakt bestand is oplopend of aflopend, maar niet tegelijk.
- Het commando SORT kent ook twee opties, en wel FOR en WHILE. Met de optie FOR kunt u records sorteren die voldoen aan een bepaalde voorwaarde (bijvoorbeeld: FOR achternaam = "Jansen"). Door gebruik van de optie WHILE kunt u een tweede, gelijktijdig geldende voorwaarde opgeven (bijvoorbeeld: WHILE Postcode = "1000 AA"). Sorteren met SORT kan in oplopende volgorde voor waarden in één veld, en tegelijk aflopend voor de waarden in een ander veld.

De letters A tot en met J en de letter M kunnen niet worden gebruikt als naam voor een database omdat deze letters zijn gereserveerd voor standaard aliasnamen. AA is echter wel een geldige bestandsnaam.

Als u op meerdere velden sorteert, moet u de belangrijkste sleutel vooraan zetten. Scheid de veldnamen met komma's.

Voorbeeld

Als u van de database Transact, in omgekeerd chronologische volgorde gesorteerd op Klant_nr, een nieuw database-bestand wilt maken met de naam Vanklant:

```
. USE Transact
. SORT ON Klant_nr/A, Dat_trans/D TO Vanklant
100% gesorteerd    12 records gesorteerd

. USE Vanklant
. LIST
```

Recordnr.	KLANT_NR	BESTEL_NR	DAT_TRANS	FACTUUR	TOTAAL_REK
1	A00005	87-113	24-03-87	.T.	125,00
2	A10025	87-116	10-04-87	.F.	1500,00
3	A10025	87-105	03-02-87	.T.	1850,00
4	B12000	87-114	30-03-87	.F.	450,00
5	C00001	87-115	01-04-87	.F.	165,00
6	C00001	87-108	23-02-87	.T.	1250,00
7	C00001	87-106	10-02-87	.T.	1200,00
8	C00002	87-110	09-03-87	.T.	175,00
9	C00002	87-107	12-02-87	.T.	1250,00
10	L00001	87-112	20-03-87	.T.	700,00
11	L00001	87-109	09-03-87	.T.	415,00
12	L00002	87-111	11-03-87	.F.	1000,00

Zie ook

COPY, INDEX

STORE

Met het commando STORE kunt u één of meer geheugenvariabelen initialiseren. Indien nodig worden ze eerst gemaakt. Tevens worden de elementen van reeksen geïnitialiseerd die eerder zijn gemaakt met het commando DECLARE.

Syntaxis

STORE <uitdrukking> TO <lijst geheugenvariabelen>/<lijst reekselementen>

Een alternatieve syntaxis voor STORE is:

<geheugenvariabele>/<reekselement> = <uitdrukking>

Standaardinstellingen

Het aantal geheugenvariabelen dat u kunt gebruiken wordt bepaald door het produkt van de systeemvariabelen MVBLKSIZE en MVMAXBLKS, die zijn opgenomen in het bestand Config.db. Als MVBLKSIZE = 50 en MVMAXBLKS = 10 (de standaardinstelling), kunt u maximaal 500 geheugenvariabelen gebruiken.

Elke reeks die u definieert neemt één *geheugenvariabelepositie* in beslag. Aan iedere reeks wordt dynamisch geheugen toegewezen om de elementen in op te slaan. Het aantal elementen van een reeks vermindert dus niet het aantal geheugenvariabeleposities dat wordt geïnitialiseerd door MVBLKSIZE en MVMAXBLKS.

Gebruik

Het gegevenstype van de variabele of het element wordt bepaald door de uitdrukking die wordt opgegeven bij STORE. Met

```
.STORE 365.25 TO N_jaar
```

maakt u een numerieke variabele met een vast decimaalteken als N_jaar nog niet bestaat. Alle numerieke variabelen die worden gemaakt van constanten als 365.25, hebben de indeling Binary Coded Decimal, tenzij met de functie FLOAT() de waarde wordt omgezet in een decimaal met een zwevend decimaalteken.

Een uitdrukking kan met één STORE-commando in meerdere geheugenvariabelen of reekselementen worden opgeslagen. U kunt elementen en geheugenvariabelen door elkaar gebruiken in dezelfde STORE-opdracht. Bijvoorbeeld:

```
.STORE 1 to M_n1, M_n2, een[1,1]
```

Met de alternatieve syntaxis <geheugenvariabele>/<reekselement> = <uitdrukking>, kunt u een uitdrukking slechts in één geheugenvariabele of in één reekselement tegelijk opslaan. De uitdrukking moet rechts van het is-gelijkteken staan en de naam van de geheugenvariabele mag geen commando van dBASE IV zijn en evenmin een afkorting van zo'n commando.

Als er al een geheugenvariabele of reeks bestaat met dezelfde naam, wordt deze overschreven, tenzij de ene variabele of reeks als PUBLIC is gedefinieerd en de andere als PRIVATE.

De namen van geheugenvariabelen en reeksen mogen uit letter, cijfers en onderstrepingstekens bestaan. De naam mag 10 tekens lang zijn en moet beginnen met een letter. Namen van één teken zijn toegestaan, maar u moet de letters A tot en met J en de letter M niet gebruiken omdat die namen in conflict kunnen komen met aliasnamen. Combinaties van twee letters, zoals AA, zijn wel toegestaan.

Iedere keer dat u gegevens in een variabele opslaat, wordt het gegevenstype van die variabele ingesteld. Het type is altijd hetzelfde als het type van de gegevens die worden opgeslagen. Geheugenvariabelen hebben geen vast gegevenstype en hoeven niet een bepaald gegevenstype te blijven nadat ze zijn geïnitialiseerd.

Tips

Als een geheugenvariabele een onverwachte waarde oplevert, kan het zijn dat er al een veld of een rekenveld met dezelfde naam bestaat in het actieve database-bestand.

Een veld met dezelfde naam als een geheugenvariabele heeft een hogere prioriteit bij alle bewerkingen behalve macrosubstitutie (&). Om een geheugenvariabele de hoogste prioriteit te geven moet u *M->* (het symbool dat gebruikt wordt om een geheugenvariabele aan te geven) voor de naam van de variabele zetten. Bijvoorbeeld:

M-> <geheugenvariabele>

U kunt de namen van variabelen ook met een bepaalde letter laten beginnen, bijvoorbeeld M_, zodat ze niet worden verward met veldnamen. Hierdoor kunt u ook RELEASE en SAVE gebruiken voor een bepaalde verzameling variabelen en de rest van de variabelen ongemoeid laten. Bijvoorbeeld:

```
. RELEASE ALL LIKE M_*        && Variabelen die beginnen met "M_" worden vrijgegeven.
. SAVE TO Opslag ALL LIKE A_* && Er wordt een bestand gemaakt met de naam Opslag.mem voor alle
                                 variabelen waarvan de naam begint met "A_".
```

Voorbeelden

U kunt als volgt een logische geheugenvariabele met de naam L_betaald maken en instellen op onwaar (.F.):

```
.STORE .F. TO L_betaald
.F.
```

Op de volgende manier kunt u een binaire numerieke variabele met zwevend decimaalteken maken:

```
.F_jaar = FLOAT(365.25)
```

U kunt de geheugenvariabelen Mkosten1, MKosten2 en Mkosten3 als volgt instellen op 0:

```
.STORE 0 TO Mkosten1, Mkosten2, Mkosten3
```

In het volgende voorbeeld worden de reekselementen Maand[1,2], Maand[2,2] en Maand[3,2] ingesteld op respectievelijk januari, februari en maart:

```
.DECLARE Maand[3,2]
. Maand[1,2] = "januari"
januari
. Maand[2,2] = "februari"
februari
. Maand[3,2] = "maart"
maart
```

Er zijn verschillende manieren om datumvariabelen te maken:

```
. X = DATE()
. Y = CTOD("01-02-89")
. Nieuwmnd = {31-12-88}
```

Zie ook

&, CLEAR ALL, CLEAR MEMORY, DECLARE, LIST/DISPLAY MEMORY, PARAMETERS, PRIVATE, PUBLIC, RELEASE, RESTORE, SAVE

SUM

Met het commando SUM kunt u het totaal van numerieke uitdrukkingen berekenen en opslaan in geheugenvariabelen of reeksen.

Syntaxis

SUM [<Nuitdr lijst>] [TO <lijst geheugenvariabelen>/TO ARRAY <naam reeks>] [<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstellingen

Tenzij u een bereik of een FOR- of WHILE-clausule opgeeft, wordt het totaal van alle records berekend.

Als u geen uitdrukkingslijst opgeeft wordt het totaal van alle numerieke velden berekend.

Gebruik

Het totaal van elke numerieke uitdrukking wordt in de geheugenvariabelenlijst of reeks geplaatst waarna de resterende onderdelen van de uitdrukkingslijst worden verwerkt. Als er niet genoeg geheugenvariabelen of reekselementen zijn, worden alleen de bestaande variabelen of elementen gevuld. Als niet alle variabelen in de lijst of alle reekselementen worden gebruikt, worden de waarden van de overige variabelen of elementen niet gewijzigd.

De opgegeven reeks moet eendimensionaal zijn. De uitkomst van SUM is een numerieke variabele van het type F die afhankelijk is van de bronuitdrukking.

Voorbeeld

In het volgende voorbeeld wordt het totaal van de velden Stukprijs en Aantal in de reeks Martikelen geplaatst:

```
. USE Magaz
. DECLARE Martikelen[2]
. SUM Stukprijs, Aantal TO ARRAY Martikelen
17 records opgeteld
Stukprijs        Aantal
10505            20
. ? STR(Martikelen[2],3,0), "stuks tot op heden verkocht."
20 stuks tot op heden verkocht.
. ? "Het totaalbedrag is ƒ " + LTRIM(STR(Martikelen[1],9,2))
Het totaalbedrag is ƒ10505,00
```

Zie ook

AVERAGE, CALCULATE, COUNT, DECLARE, STORE,

SUSPEND

Met het commando SUSPEND kunt u een lopend programma onderbreken. Terwijl het programma is onderbroken kunt u commando's intypen bij de commandostip. Met RESUME wordt het programma verder uitgevoerd en met CANCEL wordt het onderbroken programma uit het geheugen verwijderd.

Syntaxis

SUSPEND

Gebruik

Een programma kan worden onderbroken door het commando SUSPEND in het programma op te nemen of door U te typen als antwoord op de aanwijzing **OPDRACHT**: in het debug-programma.

Als SET TRAP op OFF is ingesteld en u op de **Esc**-toets drukt terwijl een programma wordt uitgevoerd, verschijnt de aanwijzing **Annuleren Negeren Onderbreken**. Als u op **Esc** drukt en SET TRAP op ON is ingesteld, gaat dBASE IV naar het foutopsporingsscherm.

Als u **Onderbreken** kiest, kunt u ofwel een commandoregel wijzigen die is opgeslagen in de logboekbuffer, of nieuwe commando's uitvoeren. Commando's van programmabestanden worden niet opgeslagen in de logboekbuffer. Wanneer u de .prg wijzigt van een dBASE-programma dat onderbroken is, moet u het programma annuleren en compileren om de wijzigingen door te voeren. U kunt een programma tijdens het uitvoeren echter wel wijzigen door het vanaf het Debug-scherm met de optie Programmabestand te bewerken.

Typ RESUME wanneer u het programma verder wilt uitvoeren.

Als u geheugenvariabelen maakt terwijl het programma is onderbroken, zijn die PRIVATE voor de procedure die is onderbroken.

Met CANCEL kunt u alle DO-bestanden annuleren, ook de bestanden die zijn onderbroken. Procedurebestanden worden gesloten met CLOSE PROCEDURE.

Als u terugkeert naar het programma door DO <bestandsnaam> in te typen bij de commandostip terwijl het programma is onderbroken, kan het zijn dat u uiteindelijk geen geheugen meer over hebt. Het onderbroken programma blijft in het geheugen staan en wacht op het commando RESUME, CANCEL of QUIT. Gebruik RESUME om een programma verder uit te voeren in plaats van het programma weer vanaf het begin uit te voeren. Gebruik CANCEL om alle geopende programmabestanden te sluiten.

Zie ook

CANCEL, COMPILE, DEBUG, DO, RESUME, SET ECHO, SET PROCEDURE, SET STEP, SET TALK, SET TRAP

TEXT

Met het commando TEXT kunt u blokken tekst naar het scherm, de printer of een bestand uitvoeren. Met dit commando kunt u eenvoudig en gemakkelijk tekst naar een uitvoereenheid schrijven.

Syntaxis

TEXT
<teksttekens>
ENDTEXT

Gebruik

De teksttekens worden in precies dezelfde volgorde uitgevoerd als waarin ze oorspronkelijk zijn ingevoerd in een programma.

Tekstuitvoer gaat rechtstreeks naar een actieve uitvoereenheid, bijvoorbeeld het scherm, de printer of een ander bestand.

De macrofuncties (&) die zijn opgenomen in de tekst worden niet uitgebreid, maar worden net zo uitgevoerd als ze zijn ingetypt.

Wanneer SET PRINTER op ON is ingesteld wordt de uitvoer naar de printer gezonden.

Voorbeeld

De volgende twee zinnen tekst worden weergegeven als dit TEXT...ENDTEXT-blok in een programma is opgenomen:

```
TEXT
Dit is een voorbeeld van tekst die wordt weergegeven met het commando
TEXT. Deze tekst wordt direct naar het scherm gezonden, zonder dat dBASE IV de tekst interpreteert.
ENDTEXT
```

Zie ook

@, ?/??, ???, LIST/DISPLAY, SET PRINTER

TOTAL

Met het commando TOTAL kunt u de som berekenen van de numerieke velden in het actieve database-bestand. Er wordt een nieuw bestand gemaakt om de uitkomsten in op te slaan. In de numerieke velden in het TO-database-bestand komen de uitkomsten te staan van alle records met dezelfde sleutelwaarde als de oorspronkelijke database.

Syntaxis

TOTAL ON <sleutelveld> TO <bestandsnaam> [FIELDS <veldenlijst>]
[<bereik>] [FOR <voorwaarde>] [WHILE <voorwaarde>]

Standaardinstellingen

De bestandsnaam die wordt opgegeven bij TO moet een stationsaanduiding bevatten als u niet wilt dat het bestand op het standaardstation wordt opgeslagen. dBASE IV gebruikt de extensie .dbf, tenzij u iets anders opgeeft.

Het totaal van alle records wordt berekend, tenzij anders wordt bepaald door het bereik of een FOR- of WHILE-clausule. De som van alle numerieke velden in het record wordt berekend, tenzij anders wordt bepaald door de FIELDS <veldenlijst>.

Als een catalogus wordt gebruikt, wordt het TO-bestand aan de catalogus toegevoegd.

Gebruik

Het actieve database-bestand moet zijn geïndexeerd of gesorteerd op het sleutelveld. Als de bestandsnaam van TO al bestaat, wordt eerst een waarschuwing gegeven voordat het bestand wordt overschreven, tenzij SET SAFETY op OFF is ingesteld.

Alle records met hetzelfde sleutelveld worden één record in de TO-database. Elk numeriek veld in de FIELDS <veldenlijst> bevat een totaal. Alle andere velden bevatten de gegevens die in het eerste record staan van de verzameling records met hetzelfde sleutelveld.

De structuur van het TO-bestand is hetzelfde als de structuur van het actieve database-bestand, behalve dat memovelden niet naar het nieuwe bestand worden gekopieerd. Numerieke velden van het type N in het bronbestand worden numerieke totalen van het type N in het TO-bestand en velden van het type F worden totalen van het type F.

Tips

De letters A tot en met J en M mogen niet worden gebruikt als namen voor database-bestanden, omdat die zijn gereserveerd voor standaard aliasnamen. AA is wel een geldige bestandsnaam.

Als een veld in het TO-bestand niet groot genoeg is voor het totaal, plaatst dBASE IV asterisken in het veld (om overloop aan te geven). U kunt dit voorkomen door MODIFY STRUCTURE te gebruiken om de velden in het actieve database-bestand zo groot te maken dat het grootste totaal erin kan worden opgeslagen.

Voorbeeld

U kunt als volgt van het database-bestand Transact een nieuw database-bestand maken met de naam Bedragen, waarin de som van Totaal_rek wordt opgeslagen voor elk Klant_nr:

```
.USE Transact ORDER Klant_nr
Hoofdindex: KLANT_NR

.TOTAL ON Klant_nr TO Bedragen
                    12 records in totaal
                    7 records gegenereerd
.USE Bedragen

.LIST Klant_nr, Totaal_rek
Recordnr.    KLANT_NR      TOTAAL_REK
     1       A00005         125,00
     2       A10025        3350,00
     3       B12000         450,00
     4       C00001        2615,00
     5       C00002        1425,00
     6       L00001        1115,00
     7       L00002        1000,00
```

Zie ook

INDEX, MODIFY STRUCTURE, SET SAFETY, SORT

TYPE

Met TYPE kunt u de inhoud van een ASCII-tekstbestand weergegeven.

Syntaxis

TYPE <bestandsnaam> [TO PRINTER/TO FILE <bestandsnaam>] [NUMBER]

Standaardinstellingen

Als het bestand niet op het actieve station staat, moet een stationsaanduiding worden opgegeven. Als SET PATH niet wijst naar de plaats van het bestand, moeten de stationsaanduiding en het pad worden opgenomen in de bestandsnaam.

Bestanden met uitgebreide ASCII-tekens worden weergegeven in tekens uit de uitgebreide IBM ASCII-set.

Gebruik

Met TYPE kunnen alleen standaard ASCII-tekstbestanden worden weergegeven, zoals commando- of procedurebestanden. Database-bestanden, indexbestanden en databasetekstbestanden die worden weergegeven met TYPE zijn meestal niet goed te lezen omdat ze ook informatie bevatten die wordt gebruikt door dBASE. TYPE kan niet worden gebruikt voor een geopend bestand. In een programma mag TYPE dus nooit worden gebruikt voor het programma zelf of een ander geopend bestand.

Als uw printer de uitgebreide IBM ASCII-tekenset niet ondersteunt, zoals dubbele lijnen of hoeken, worden die tekens vervangen door normale tekens, zoals het plus- en minteken.

Opties

Als u het sleutelwoord NUMBER opgeeft in het commando, worden regelnummers weergegeven of afgedrukt. De regelnummers komen overeen met de regels in het programma, niet met de regels op het scherm. Regels die doorlopen in het programma gelden als aparte regels. Als SET HEADING op ON is ingesteld, wordt ook een paginatitel weergegeven die bestaat uit de bestandsnaam, de datum en het paginanummer. De complete bestandsnaam zoals die is ingevoerd op de commandoregel wordt aan de linker kant weergeven, gevolgd door de huidige datum in SET DATE-opmaak. Paginanummers worden aan de rechter kant van de pagina weergegeven.

Als u TO PRINTER opgeeft, wordt de uitvoer naar de printer gezonden.

Als u TO FILE opgeeft, wordt de uitvoer naar een bestand gezonden en zult u gevraagd worden een bestandsnaam op te geven.

Zie ook

SET DATE, SET HEADING, SET PATH

UNLOCK

Met UNLOCK kunt u de vergrendeling van records en bestanden verwijderen, zodat andere gebruikers de gegevens kunnen wijzigen.

Syntaxis

UNLOCK [ALL/IN <alias>]

Gebruik

Als een record of bestand is vergrendeld, kan dit niet worden gewijzigd of bijgewerkt door een andere gebruiker totdat met het commando UNLOCK het record of bestand is ontgrendeld.

Als het commando UNLOCK wordt gebruikt, worden de records of het bestand in het gekozen werkgebied ontgrendeld die door de actieve gebruiker zijn vergrendeld. Als u UNLOCK ALL gebruikt worden alle vergrendelingen ongedaan gemaakt, zowel van bestanden als van records, in alle werkgebieden. Met UNLOCK IN <alias> kunt u de records of het bestand in het opgegeven werkgebied ontgrendelen. De sleutelwoorden ALL en IN heffen elkaar op.

Als de laatste vergrendeling een record betrof, worden alle records ontgrendeld door UNLOCK. Als de laatste vergrendeling een bestand betrof, wordt het bestand ontgrendeld. Bij sommige commando's wordt het bestand of record eerst automatisch vergrendeld voordat het commando wordt uitgevoerd. Het wordt weer ontgrendeld als het commando is uitgevoerd en u hoeft het commando UNLOCK dus niet te gebruiken, tenzij het commando niet helemaal is uitgevoerd.

Als u een bestand of record vergrendelt dat is gerelateerd aan andere geopende bestanden of records, worden alle records en bestanden in de relatie automatisch vergrendeld. Als u het bestand of record ontgrendelt, worden ook alle gerelateerde bestanden en records automatisch ontgrendeld.

Voorbeeld

In het volgende voorbeeld wordt de functie RLOCK() gebruikt om een record te vergrendelen en het commando UNLOCK om het te ontgrendelen:

```
. USE Afnemers
. ? RLOCK()
.T.
. REPLACE Achternaam WITH "Nelissen"
                    1 record vervangen
. REPLACE Voornaam WITH "Jan"
                    1 record vervangen

. UNLOCK
```

Zie ook

FLOCK(), RLOCK()/LOCK()

UPDATE

Met het commando UPDATE kunt u velden in het actieve database-bestand vervangen door gegevens uit een ander database-bestand. De velden worden gewijzigd doordat één sleutelveld wordt gebruikt om de records in de twee database-bestanden met elkaar te vergelijken.

Syntaxis

UPDATE ON <sleutelveld> FROM <alias>
REPLACE <veldnaam 1> WITH <uitdrukking 1>
[,<veldnaam 2> WITH <uitdrukking 2>...]
[RANDOM]

Gebruik

Het bestand dat moet worden bijgewerkt, moet het actieve bestand zijn. Het database-bestand FROM is een geopend database-bestand in een ander werkgebied, opgegeven met de alias.

Het sleutelveld moet dezelfde naam hebben in beide database-bestanden. Als het bestand dat moet worden bijgewerkt geen unieke sleutelveldwaarde heeft voor elk record dat wordt ontvangen van het FROM-bestand, wordt alleen het eerste record met dezelfde sleutelwaarde bijgewerkt. Als er in het FROM-bestand meerdere records dezelfde sleutelveldwaarde hebben, wordt het record dat in het bijgewerkte bestand dezelfde sleutelwaarde heeft telkens weer bijgewerkt en worden alleen de laatst ontvangen waarden behouden.

Opties

Beide bestanden moeten in oplopende volgorde zijn gesorteerd of geïndexeerd, tenzij RANDOM wordt gebruikt. Als u RANDOM gebruikt moet het database-bestand dat wordt bijgewerkt zijn geïndexeerd op het sleutelveld, maar mag het FROM-bestand in een willekeurige volgorde staan. Hoewel het is toegestaan, moet u het sleutelveld niet opgeven bij de optie REPLACE. Sleutelvelden mogen niet worden gewijzigd omdat ze door het commando worden vergeleken.

Als er in de uitdrukking van REPLACE een veld uit het FROM-bestand wordt gebruikt (wat meestal het geval is), moet het WITH-veld worden genoteerd als een aliasveld (->).

Voorbeeld

In dit deel van een voorbeeldprogramma wordt het veld Totaal_rek van het database-bestand Transact vervangen door het produkt van de velden Aantal en Stukprijs van het database-bestand Magaz:

```
USE Transact ORDER Bestel_nr
REPLACE ALL Totaal_rek WITH 0
USE Magaz ORDER Bestel_nr IN 2
UPDATE ON Bestel_nr FROM Magaz REPLACE Totaal_rek WITH Magaz->Aantal *;
    Magaz->Stukprijs
LIST
```

De uitvoer van het programma ziet er als volgt uit:

Recordnr.	KLANT_NR	BESTEL_NR	DAT_TRANS	FACTUUR	TOTAAL_REK
1	A10025	87-105	03-02-87	.T.	650,00
2	C00001	87-106	10-02-87	.T.	1200,00
3	C00002	87-107	12-02-87	.T.	1250,00
4	C00001	87-108	23-02-87	.T.	1250,00
5	L00001	87-109	09-03-87	.T.	165,00
6	C00002	87-110	09-03-87	.T.	100,00
7	L00002	87-111	11-03-87	.F.	1000,00
8	L00001	87-112	20-03-87	.T.	700,00
9	A00005	87-113	24-03-87	.T.	125,00
10	B12000	87-114	30-03-87	.F.	450,00
11	C00001	87-115	01-04-87	.F.	165,00
12	A10025	87-116	10-04-87	.F.	1000,00

Zie ook

INDEX, JOIN, REPLACE, SORT, TOTAL

Met USE wordt een bestaand database-bestand geopend. Eventueel worden .ndx- en/of .mdx-indexbestanden geopend. Als het database-bestand memovelden bevat, wordt het bijbehorende .dbt-bestand automatisch geopend.

Syntaxis

USE [<bestandsnaam>/?] [IN <nummer werkgebied>]
[[INDEX <lijst .ndx- of .mdx-bestanden>]
[ORDER <naam .ndx-bestand>/[TAG]<.mdx-label>
[OF <naam .mdx-bestand>]]
[ALIAS <alias>] [EXCLUSIVE] [NOUPDATE]]

Standaardinstellingen

Tenzij IN <werkgebiednummer> wordt opgegeven, wordt het bestand in het actieve werkgebied geopend.

Met USE ↵ kunt u alle database-bestanden en bijbehorende indexbestanden in het actieve werkgebied sluiten. Met USE IN <nummer werkgebied> kunt u alle bestanden in het opgegeven werkgebied sluiten.

Tenzij u een andere extensie opgeeft, gebruikt dBASE IV de extensie .dbf voor de database-bestandsnaam en de extensie .mdx of .ndx voor de indexen die worden opgegeven in een indexlijst.

Als u informatie over een database-bestand in de catalogus wilt opnemen, moet u de catalogus openen met SET CATALOG TO of door SET CATALOG op ON in te stellen. Daarna moet u het databestand met USE openen.

Gebruik

Als er een catalogus is geopend, kunt u met USE ? een menu weergeven waarin alle gecatalogiseerde .dbf-bestanden worden weergegeven. Als er geen catalogus is geopend, worden alle .dbf-bestanden in de actieve bestandsindex weergegeven.

U kunt voor <bestandsnaam> een indirecte verwijzing gebruiken. Een indirecte verwijzing is een tekstuitdrukking die naar een bestandsnaam verwijst. Deze tekstuitdrukking kunt u altijd in plaats van een bestandsnaam gebruiken. U moet in de tekstuitdrukking een operator (gewoonlijk haakjes) gebruiken, zodat dBase weet dat de tekenreeks een uitdrukking is en niet de letterlijke bestandsnaam. Een indirecte verwijzing komt in gebruik overeen met het macrosubstitutieteken, maar voert de handeling sneller uit.

De clausule IN <werkgebiednummer> kan worden gebruikt om bestanden te openen in andere werkgebieden. <nummer werkgebied> moet een cijfer zijn tussen 1 en 10 of een van de letters A tot en met J. U kunt ook numerieke uitdrukkingen gebruiken die een waarde opleveren tussen 1 en 10, of een tekenuitdrukking die een van de letters van A tot en met J oplevert.

De volgorde waarin de records van het database-bestand worden gepresenteerd wordt bepaald door de index. Indexbestanden met de extensie .ndx zijn compatibel met dBASE III en dBASE III PLUS. Deze bestanden bevatten een verzameling wijzers naar het database-bestand die ervoor zorgen dat, wanneer de index wordt gebruikt, de records in geïndexeerde volgorde worden gepresenteerd. Het verdient aanbeveling alle dBASE III PLUS-bestanden in dBASE IV opnieuw in een index op te nemen.

In de indexbestanden met de extensie .mdx kunnen 47 labels staan. Elke label in het .mdx-bestand is een index en werkt als een .ndx-index. Als u een .mdx-bestand opent, wordt slechts één DOS-bestand gebruikt, maar hebt u 47 indexen tot uw beschikking. Als u een .ndx-bestand opent, wordt er één DOS-bestand gebruikt en kunt u slechts één index gebruiken.

Alle geopende indexen, ook .mdx-labels, worden bijgewerkt als het bijbehorende database-bestand wordt gewijzigd. Voor elk database-bestand mogen tien .ndx-bestanden zijn geopend, plus het bijbehorende .mdx-bestand.

Elk .ndx- en .mdx-bestand gebruikt een DOS-bestand. Er mogen 99 bestanden zijn geopend, tenzij een ander aantal is opgegeven bij FILES in het bestand CONFIG.SYS of Config.db (het laagste aantal is bepalend). Het beschikbare geheugen speelt ook een rol. dBASE IV gebruikt vijf DOS-bestanden. Er blijven er derhalve maximaal 94 over.

Opties

Indien het database-bestand een bijbehorend .mdx-bestand heeft, worden door USE zowel het database-bestand als het .mdx-bestand geopend. Wanneer u een ander .mdx- of .ndx-bestand wilt openenen, moet u de INDEX-clausule gebruiken, gevolgd door een bestandsnaam. Als u in de INDEX-clausule geen extensie opgeeft bij de bestandsnaam zal dBASE IV eerst zoeken naar een .mdx-bestand met de opgegeven naam. Het is daarom altijd nodig, wanneer u over .ndx- en een .mdx-bestand met dezelfde naam beschikt, de .ndx-extensie met de naam van het te openen bestand op te geven.

Als u het database-bestand hebt geopend met USE, kunt u alle indexbestanden en labels openen en sluiten met SET INDEX. U kunt een .ndx-bestand ook sluiten met CLOSE INDEX en DELETE TAG.

Met de clausule ORDER kunt u opgeven welke index de indexvolgorde bepaalt. De index die u opgeeft kan een .ndx-bestand zijn of een label in een multi-indexbestand (.mdx). De actieve index bevat de indexsleuteluitdrukking die wordt gebruikt door de commando's FIND en SEEK. Als er slechts één index wordt opgegeven, bepaalt die index de indexvolgorde en hoeft de clausule ORDER niet te worden gebruikt. De parameter <.ndx-bestandsnaam> van de clausule ORDER wordt gebruikt als er meer dan een .ndx-bestand aan het database-bestand gerelateerd is.

Met de clausule OF <naam .mdx-bestand> kunt u opgeven in welk .mdx-bestand de label is opgeslagen. Als een label niet in het .mdx-bestand staat moet OF worden gebruikt om de juiste index aan te geven.

Met het commando SET ORDER kunt u van index veranderen zonder .ndx- en .mdx-bestanden te sluiten en te openen.

Met ALIAS kan een aliasnaam worden opgeven die later kan worden gebruikt om naar een database-bestand te verwijzen. Als geen aliasnaam wordt opgegeven, wordt de bestandsnaam gebruikt als aliasnaam. U kunt ook de letters A tot en met J en de cijfers 1 tot en met 10 gebruiken om naar de tien werkgebieden te verwijzen. Als u een nieuw werkgebied kiest, wordt het database-bestand in dat werkgebied het actieve database-bestand.

Aliasnamen mogen alleen bestaan uit letters, cijfers en onderstrepingstekens (_). De naam moet beginnen met een letter. Deze beperkingen gelden niet voor bestandsnamen en het kan dus voorkomen dat een bestandsnaam niet als alias kan worden gebruikt.

Met EXCLUSIVE wordt een bestand geopend voor exlusief gebruik in een netwerk. Andere gebruikers kunnen het bestand dan niet gebruiken tot het bestand is gesloten met het commando CLEAR ALL, CLOSE of USE. Wanneer dBASE IV niet in een netwerk wordt gebruikt, worden alle bestanden als EXCLUSIVE geopend en wordt dit sleutelwoord genegeerd.

Als u de optie NOUPDATE opgeeft, mogen de gegevens in het bestand niet worden gewijzigd. Er mag dus ook niets worden toegevoegd of verwijderd. Het bestand krijgt als het ware een schrijfbeveiliging.

Recordwijzer

Als u geen indexbestand gebruikt voor een database-bestand, wordt de recordwijzer op het eerste record van het bestand geplaatst.

Als u één of meer indexen gebruikt voor een database-bestand, wordt de recordwijzer op het eerste logische record van de actieve index geplaatst. De recordwijzer volgt die index tot een andere index wordt opgeroepen met SET ORDER of SET INDEX.

Speciale gevallen

Tenzij u een alias opgeeft met het sleutelwoord ALIAS, gebruikt dBASE IV de bestandsnaam als aliasnaam. Sommige bestandsnamen, bijvoorbeeld X-y, kunnen niet worden gebruikt als aliasnaam omdat ze tekens bevatten die door dBASE voor andere doeleinden worden gereserveerd. Als er in een bestandsnaam tekens staan die niet mogen worden gebruikt in een aliasnaam, gebruikt dBASE een van de letters A tot en met J als standaard aliasnaam.

Wanneer SET TALK op ON is ingesteld, de optie ALIAS niet is opgegeven en de bestandsnaam niet kan worden gebruikt als alias, wordt de alias van het werkgebied gebruikt en verschijnt de melding **Standaard-alias is A** (of de juiste letter van het werkgebied).

Voorbeeld

In het volgende voorbeeld worden de database-bestanden Afnemers, Transact en Magaz geopend in respectievelijk werkgebied 1, 2 en 3:

```
. CLOSE ALL

. USE Afnemers INDEX Klntnaam
Hoofdindex: KLNTNAAM

. USE Transact ORDER Klant_nr IN 2
Hoofdindex: KLANT_NR

. USE Magaz ORDER Bestel_nr IN 3
Hoofdindex: BESTEL_NR
```

In het volgende voorbeeld worden zowel een .ndx- als een .mdx-bestand geopend:

```
. USE Vrienden INDEX Code ORDER Naam
```

In het volgende voorbeeld wordt een niet-bijbehorend .mdx-bestand geopend:

```
. USE Telefoon ORDER Laatste OF Tela
```

Zie ook

&(macro), CLEAR ALL, CLOSE, COPY INDEX, COPY TAG, DBF(), DELETE TAG, INDEX, SELECT, SET CATALOG, SET EXCLUSIVE, SET INDEX, SET ORDER

WAIT

Met WAIT worden alle activiteiten van dBASE IV gestaakt tot er op een toets wordt gedrukt.

Syntaxis

WAIT [<aanwijzing>] [TO <geheugenvariabele>]

Standaardinstellingen

De geheugenvariabele krijgt de waarde nul (ASCII-waarde 0) als u op ↵ drukt. Als u op een ander niet-afdrukbaar teken drukt, wordt het grafische teken met dezelfde ASCII-waarde opgeslagen in de geheugenvariabele.

Gebruik

De aanwijzing is een tekenuitdrukking. Als u geen aanwijzing opgeeft wordt de melding **Druk op een toets om door te gaan** weergegeven.

Wanneer u op **Esc** drukt terwijl dBASE onderbroken is doordat u het commando WAIT hebt gegeven en SET ESCAPE op ON is ingesteld, ontvangt u een onderbrekingsfoutmelding. U moet daarom SET ESCAPE op OFF instellen voordat u het commando WAIT geeft.

Opties

Als u de optie TO <geheugenvariabele> opgeeft, wordt een geheugenvariabele van het type teken gemaakt waarin de ingedrukte toets wordt opgeslagen.

Voorbeeld

U kunt WAIT gebruiken om een programma van dBASE tijdelijk te onderbreken. Neem daarvoor de volgende commando's op in uw programma:

```
WAIT "Verder gaan? (J/N) " TO M_ver
IF UPPER(M_ver) = "N"
    RETURN
ENDIF
```

Zie ook

ACCEPT, INPUT, ON KEY, STORE

ZAP

Met ZAP kunt u alle records uit het actieve database-bestand verwijderen.

Syntaxis

ZAP

Gebruik

ZAP is hetzelfde als het commando DELETE ALL gevolgd door PACK, maar werkt sneller.

Als SET SAFETY op ON is ingesteld, reageert dBASE IV met de melding **Opschonen <bestandsnaam>? (J/N)**.

Alle geopende indexbestanden in het actieve werkgebied worden automatisch bijgewerkt.

dBASE IV neemt de ruimte van het verwijderde bestand weer in beslag, ook de ruimte die in beslag werd genomen door memo- en indexbestanden.

Speciale gevallen

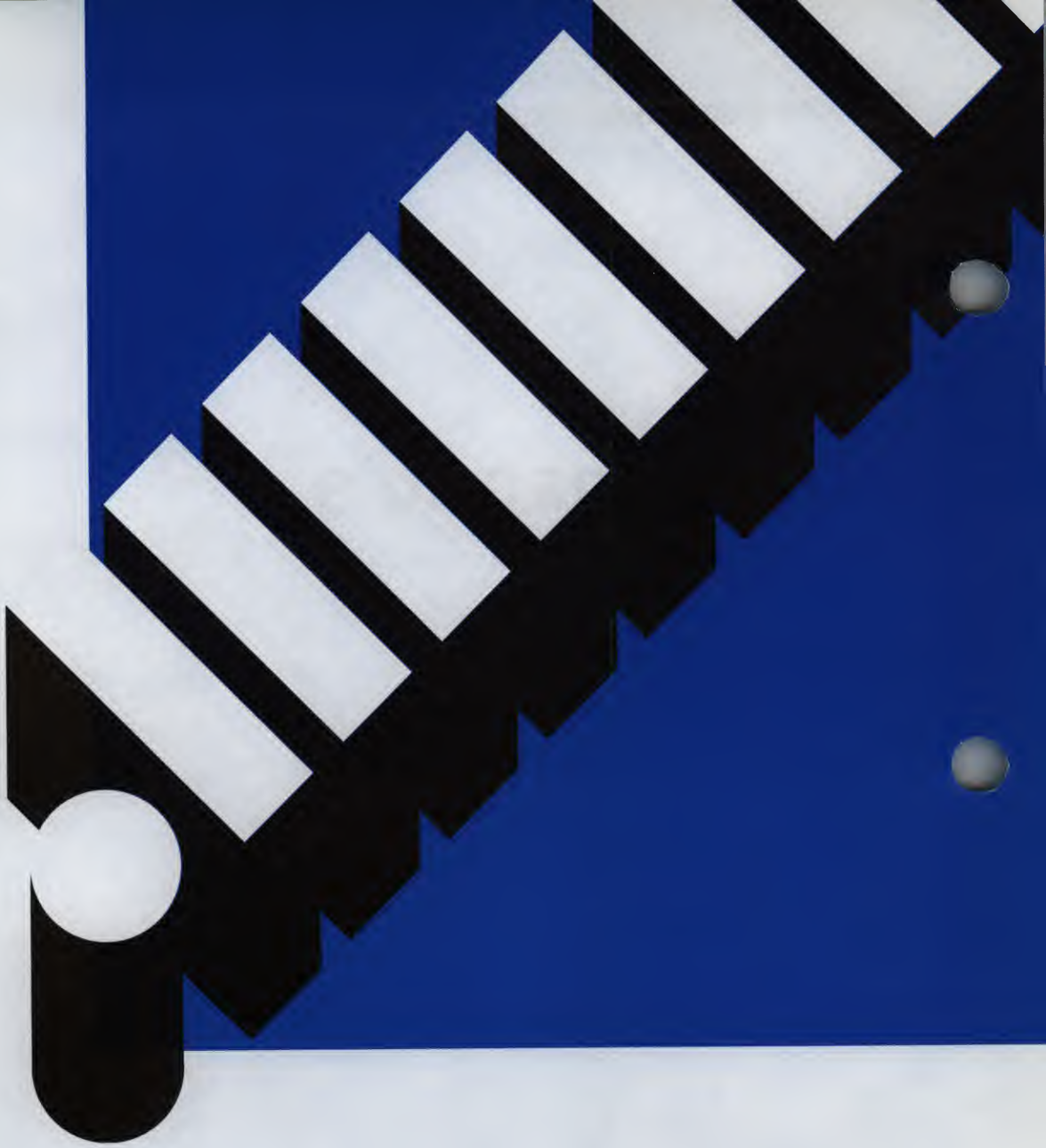
In een netwerkomgeving kunt u het commando ZAP pas dan opgeven, als u alleen met het database-bestand werkt.

Zie ook

DELETE, PACK, SET SAFETY

Naslaghandboek van dBASE IV

Set-Commando's



i 1 2 3 4 5 6 7 A B

C D E F G H In

SET-commando's

SET

SET is een menugestuurd commando waarmee de waarden van veel SET-commando's kunnen worden gewijzigd.

Syntaxis

SET

Gebruik

Met het commando SET kunt u vanaf de commandostip het menu **SET** weergeven. De menubalk bovenaan het scherm laat vijf opties zien.

Het submenu **Opties** geeft de standaardinstellingen weer van veel SET-commando's. Vanuit dit menu kunt u de instellingen wijzigen.

Met het submenu **Weergave** kunt u de schermattributen voor kleur en intensiteit wijzigen voor tekst, kopteksten, de statusregel en velden. In een ander venster is te zien hoe uw keuze op het scherm wordt weergegeven. Als u tevreden bent met de weergave drukt u op **Ctrl-End**. Op die manier blijven de wijzigingen die u hebt aangebracht van kracht gedurende de rest van de actieve sessie in dBASE IV. Als u op **Esc** drukt, worden de wijzigingen niet opgeslagen. Wanneer u dBASE IV verlaat worden alle waarden, behalve die van de weergavemodus, weer ingesteld op de standaardwaarde.

Met het submenu **Toetsen** kunt u bepaalde functietoetsen en toetscombinaties programmeren.

Met het submenu **Station** kunt u het standaardstation, het zoekpad en de werkeenheid/directory wijzigen.

Met het submenu **Bestanden** kunt u ALTERNATE-bestanden, indelingsbestanden, uitvoerbestanden en indexbestanden maken.

Zie ook

Raadpleeg *Menu's van dBASE IV* voor een volledige bespreking van het menu SET.

U kunt ook de standaardinstelling in het bestand Config.db wijzigen. Raadpleeg *Aan de slag met dBASE IV* voor meer informatie over het bestand Config.db.

SET ALTERNATE

Met SET ALTERNATE wordt alle uitvoer opgenomen in een tekstbestand, behalve de uitvoer van schermgrote commando's.

Syntaxis

SET ALTERNATE on/OFF

SET ALTERNATE TO [<bestandsnaam> [ADDITIVE]]

Standaardinstelling

De standaardinstelling van SET ALTERNATE is OFF.

Het ALTERNATE-bestand wordt opgeslagen in het standaard station tenzij u een stationaanduidig in de bestandsnaam opneemt. Het bestand krijgt de extensie .txt, tenzij een andere extensie wordt opgegeven.

Met SET ALTERNATE TO ↵ kunt u een geopend ALTERNATE-bestand sluiten. Dat kan ook met CLOSE ALTERNATE.

Gebruik

Dit commando bestaat uit twee delen:

Met SET ALTERNATE TO <bestandsnaam> wordt een tekstbestand gemaakt en geopend. Een eventueel bestand met dezelfde naam wordt overschreven.

Als u de optie ADDITIVE gebruikt, wordt de cursor aan het einde van het ALTERNATE-bestand geplaatst voordat de uitvoer naar het bestand wordt weggeschreven. Op die manier kan informatie aan een bestand worden toegevoegd zonder dat dit wordt overschreven.

Als u het commando SET ALTERNATE ON geeft, wordt begonnen met het opnemen van de toesenbordinvoer en de beeldschermweergave in het ALTERNATE-bestand. Schermgrote bewerkingen als @...SAY, EDIT, BROWSE en APPEND worden niet opgenomen.

Met SET ALTERNATE OFF stopt u met schrijven naar het tekstbestand zonder het tekstbestand te sluiten. U kunt de opname weer starten met SET ALTERNATE ON. Het tekstbestand wordt niet gesloten totdat het comando CLOSE ALTERNATE of SET ALTERNATE TO ↵ wordt gegeven. Geef altijd het commando CLOSE ALTERNATE voordat u het tekstbestand gebruikt. Het tekstbestand dat wordt gemaakt als u het commando SET ALTERNATE geeft, is een standaard ASCII-tekstbestand dat kan worden bewerkt met MODIFY COMMAND of met een tekstverwerker.

Als u niet wilt dat het ALTERNATE-bestand begint met een blanco regel, moet u het eerste ? wijzigen in ??. Het commando ? verzendt altijd een terugloop en een regeldoorvoer voordat de tekst wordt afgedrukt.

Voorbeelden

De volgende voorbeelden laten zien hoe de commando's van SET ALTERNATE functioneren. Om een ALTERNATE-tekstbestand te maken typt u:

```
. SET ALTERNATE TO Tkstbest
```

De commando's die u geeft nadat u een alternate-bestand hebt geopend, worden pas opgenomen nadat u het commando SET ALTERNATE ON hebt gegeven. Typ:

```
. SET ALTERNATE ON
```

Alle commando's van dBASE IV die worden uitvoerd, worden opgeslagen in het ALTERNATE-bestand tot u het ALTERNATE-bestand sluit of het opnemen van commando's onderbreekt.

U kunt het opnemen van commando's als volgt onderbreken:

```
. SET ALTERNATE OFF
```

Met het volgende commando begint u weer met opnemen:

```
. SET ALTERNATE ON
```

U kunt het tekstbestand als volgt sluiten:

```
. CLOSE ALTERNATE
```

Als voorbeeld van de optie ADDITIVE maakt u met het commando MODIFY FILE een bestand met de naam Alt_uit.txt. Typ de volgende tekst:

Jantje zag eens pruimen hangen.

Sla het bestand op. Neem nu de volgende commando's op in een programmabestand:

```
SET ALTERNATE TO Alt_uit.txt ADDITIVE
SET ALTERNATE ON
?? "Oh, als eieren zo groot"
CLOSE ALTERNATE
```

In het bestand Alt_uit.txt staat nu:

```
Jantje zag eens pruimen hangen.
Oh, als eieren zo groot.
```

Zie ook

CLOSE, SET()

SET AUTOSAVE

Met het commando SET AUTOSAVE wordt een record na een I/O-bewerking opgeslagen.

Syntaxis

SET AUTOSAVE on/OFF

Standaardinstelling

De standaardinstelling van SET AUTOSAVE is OFF.

Gebruik

Met het commando SET AUTOSAVE verkleint u de kans op verlies van gegevens, omdat de records dan worden opgeslagen nadat deze zijn gewijzigd. Wanneer SET AUTOSAVE op OFF is ingesteld (de standaardinstelling), worden de records door dBASE IV op schijf opgeslagen zodra de recordbuffer vol is.

De commando's waarop SET AUTOSAVE betrekking heeft zijn APPEND, APPEND FROM, BROWSE, PACK, READ, REINDEX, REPLACE en ZAP. Wanneer SET AUTOSAVE op ON is ingesteld, vervangt u met REPLACE elk record zodra het is gewijzigd, ook als het bereik meer dan een record omvat.

Als het database-bestand een memoveld bevat, wordt met SET AUTOSAVE tegelijkertijd ook het .dbt-bestand bijgewerkt.

U kunt SET AUTOSAVE op ON instellen vanaf de commandostip of ON kiezen in het menu SET. U kunt SET AUTOSAVE ON ook opnemen in het bestand Config.db.

Zie ook

SET(),

Aan de slag met dBASE IV

SET BELL

Met het commando SET BELL kunt u het geluidssignaal instellen dat u waarschuwt voor het einde van het veld of voor een foutieve of ongeldige ingave. Met SET BELL kunt u niet alleen het geluidssignaal in- en uitschakelen, maar ook de frequentie en tijdsduur ervan instellen.

Syntaxis

SET BELL ON/off

SET BELL TO [<frequentie>,<duur>]

Standaardinstelling

De standaardinstelling van SET BELL is ON. De standaardfrequentie is 512 hertz en de duur is ingesteld op 2 tikken. Een tik is ongeveer 0,0549 seconden.

Gebruik

U kunt frequenties gebruiken tussen de 19 en 10.000 cycli per seconde. Voor een lager geluid moet u frequenties tussen de 20 en 550 instellen. Voor een hoger geluid moet u frequenties tussen de 550 en 5.500 instellen. De tijdsduur kan 2 tot 19 tikken zijn. De wijziging wordt direct van kracht en blijft actief gedurende de rest van de sessie. U kunt de instelling van BELL wijzigen in het bestand Config.db, zodat u automatisch uw eigen instellingen voor het geluidssignaal gebruikt als u dBASE IV start. Raadpleeg *Aan de slag met dBASE IV* voor meer informatie.

Typ het volgende commando om de standaardinstelling weer te gebruiken:

```
. SET BELL TO ↵
```

Voorbeelden

Typ het volgende commando om de frequentie en duur van het geluidssignaal in te stellen vanaf de commandostip:

```
. SET BELL TO 20,1
. ?CHR(7)
```

of:

```
. SET BELL TO 500,12
. ?CHR(7)
```

Zie ook

SET()

SET BLOCKSIZE

Met het commando SET BLOCKSIZE kunt u de standaard blokgrootte van memovelden en samengestelde indexbestanden (.mdx) wijzigen.

Syntaxis

SET BLOCKSIZE TO <Nuitdr>

Gebruik

<Nuitdr> is een numeriek argument met een waarde tussen 1 en 32. De standaard blokgrootte van memovelden wordt gewijzigd door het argument met 512 te vermenigvuldigen om de werkelijke blokgrootte in bytes te berekenen. De standaard blokgrootte is 1. Dat is de enige grootte die compatibel is met dBASE III PLUS.

Nadat de blokgrootte is gewijzigd, gebruiken memovelden die worden gemaakt met CREATE/MODIFY STRUCTURE en COPY de nieuwe blokgrootte.

Met SET BLOCKSIZE kunt u de blokgrootte wijzigen als een bestaand memobestand een inefficiënte blokgrootte gebruikt. Kopieer het database-bestand daarna naar een nieuw bestand. Het nieuwe bestand gebruikt de nieuwe blokgrootte.

Tips

In grote blokken kunnen tekenreeksen sneller worden bewerkt, maar I/O-bewerkingen zijn meestal langzamer. In kleine blokken worden tekenreeksen langzamer bewerkt, maar zijn de prestaties over het algemeen beter.

Als u vaak grote memovelden gebruikt, kunt u de blokgrootte wijzigen in 7 of 8. Als u te grote blokken gebruikt wordt de verwerkingssnelheid lager en worden de .dbt-bestanden groter.

U kunt ook de parameter BLOCKSIZE in het bestand Config.db wijzigen. Raadpleeg daarvoor *Aan de slag met dBASE IV*.

Zie ook

COPY, COPY INDEXES, CREATE/MODIFY STRUCTURE, INDEX, SET()

SET BORDER

Met het commando SET BORDER kunt u de randen van menu's, vensters en @-commando's wijzigen van een enkele lijn in een aantal, door de gebruiker gedefinieerde, speciale randtekens.

Syntaxis

SET BORDER TO [SINGLE/DOUBLE/PANEL/NONE/<lijnspecificatie>]

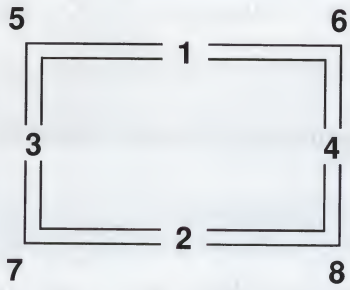
De lijnspecificatie kan als volgt worden beschreven:

[<1>] [,[<2>] [,[<3>] [,[<4>] [,[<5>]
[,[<6>] [,[<7>] [,[<8>]]]]]]]]

waarin de cijfers staan voor:

- de decimale waarde van een willekeurig IBM ASCII-teken
- een tekenreeks
- een geheugenvariabele
- een CHR()-functie

In afbeelding 3-1 staat de volgorde waarin de hoeken en zijden van het kader moeten worden opgegeven.



Afbeelding 3-1 Volgorde van kadertekens

SET BORDER

Standaardinstelling

Standaard worden de randen getekend met een enkele lijn.

Gebruik

Met SET BORDER worden de randen van objecten die al zijn gedefinieerd, niet gewijzigd. Het commando SET BORDER sorteert pas effect als u het gebruikt voordat u een menu, popup of venster definieert.

Met het commando SET BORDER kunt u de standaardrand wijzigen in een kader met dubbele lijnen, een paneel met omgekeerde weergave, een ander ASCII-teken of u kunt de rand weglaten.

Met SET BORDER TO DOUBLE wijzigt u de rand in een kader met dubbele lijnen.

Met SET BORDER TO ↵ herstelt u het standaard kader met enkele lijnen.

Met SET BORDER TO PANEL wijzigt u de rand in een kader met omgekeerde weergave, net alsof ASCII 219 is gebruikt.

De instelling van de randen heeft ook invloed op de commando's @...SAY en @...TO en op de randen van vensters. De @-commando's tekenen lijnen met de standaardinstelling van SET BORDER.

Kaders die expliciet zijn gedefinieerd met het commando DEFINE WINDOW, hebben een hogere prioriteit dan de instellingen van SET BORDER.

Voorbeelden

Om een rand met asterisken om een gedeelte van het scherm te maken moet u het volgende commando gebruiken:

```
. SET BORDER TO "*"
. @10, 15 TO 20, 55
```

Om de randen zo in te stellen dat de boven- en onderzijde worden getekend met dubbele lijnen, de linker en rechter zijde met enkele lijnen en de hoeken ook met enkele lijnen, moet u de volgende commando's gebruiken:

```
. SET BORDER TO DOUBLE
. SET BORDER TO ,,179,179,213,184,212,190
```

Eerst wordt de rand ingesteld op dubbele lijnen. Daarna wordt een nieuwe rand gedefinieerd, waarbij ASCII-waarden worden gebruikt om de lijntekens weer te geven van de uitgebreide tekenset van IBM. De twee komma's zorgen ervoor dat de dubbele lijnen voor de boven- en onderkant niet worden gewijzigd. Met 179 wordt de linker en rechter zijde ingesteld op een enkele lijn. De hoeken zijn een combinatie van dubbele en enkele lijnen. In plaats van twee commando's kunt u ook één commando gebruiken:

```
. SET BORDER TO 205,205,179,179,213,184,212,190
```

of

```
SET BORDER TO 205,205, chr(179), chr(179), "+", "+", "+", "+"
```

Als u de rand wilt wijzigen in een balk met omgekeerde weergave:

```
. SET BORDER TO PANEL
```

Zie ook

@, @...TO, DEFINE WINDOW

SET CARRY

Met het commando SET CARRY kunt u gegevens uit een voorafgaand record naar een volgend record kopiëren. Daarbij worden alle velden bijgewerkt of alleen de opgegeven velden, afhankelijk van de gebruikte syntaxis.

Syntaxis

SET CARRY on/OFF

SET CARRY TO [<veldenlijst> [ADDITIVE]]

Standaardinstelling

De standaardinstelling van SET CARRY is OFF.

Dit commando heeft geen gevolgen voor INSERT BLANK of APPEND BLANK. Er wordt nog steeds een leeg record toegevoegd.

Gebruik

Wanneer SET CARRY op ON is ingesteld, worden tijdens de bewerkingen APPEND en INSERT alle velden naar het volgende record gekopieerd.

Als u het commando SET CARRY TO <veldenlijst> geeft, worden tijdens de bewerkingen APPEND, BROWSE, EDIT (wanneer u nieuwe records toevoegt) en INSERT alleen de opgegeven velden gekopieerd. De veldenlijst die is opgegeven bij SET FIELDS heeft een hogere prioriteit dan de veldenlijst van SET CARRY. Als u een veldenlijst opgeeft met SET FIELDS of BROWSE, worden alleen de hierbij opgegeven velden gekopieerd.

Als u de optie ADDITIVE opgeeft, worden de velden in de veldenlijst toegevoegd aan de veldenlijst die eerder is opgegeven met SET CARRY.

Als u geen veldenlijst opgeeft en de syntaxis SET CARRY TO ↵ gebruikt, wordt de standaardinstelling weer van kracht en worden alle velden bij de bewerking betrokken.

Als SET CARRY op OFF is ingesteld, worden de velden niet bijgewerkt. Als u het commando SET CARRY TO <veldenlijst> geeft, wordt SET CARRY automatisch op ON ingesteld, zodat de veldenlijst bij de bewerking kan worden betrokken.

OPMERKING
Wanneer er een index actief is, werkt het commando INSERT net zo als het commando APPEND. Stel dat u een database-bestand hebt met tien records en dat u een nieuw record (nummer 6) wilt toevoegen dat dezelfde gegevens bevat als record 5. U herstelt in dat geval eerst de natuurlijke volgorde van de database met SET ORDER TO, waarna u SET CARRY op ON instelt en met INSERT een record toevoegt.

Zie ook

APPEND, BROWSE, EDIT, INSERT, READ, SET(), SET FIELDS, SET FORMAT

SET CATALOG

Met het commando SET CATALOG kunt u een catalogusbestand maken of openen. Wanneer een catalogus is geopend en SET CATALOG op ON is ingesteld, worden nieuwe bestanden die u gebruikt of maakt, toegevoegd aan de catalogus.

Syntaxis

SET CATALOG on/OFF

SET CATALOG TO [<bestandsnaam>/?]

Standaardinstelling

De standaardinstelling van SET CATALOG is OFF, waardoor er geen catalogi worden gekozen. Wanneer u echter een catalogus kiest met SET CATALOG TO wordt SET CATALOG automatisch op ON ingesteld.

Gebruik

Met SET CATALOG TO <bestandsnaam> kunt u een bestaande catalogus openen of, als er geen catalogus bestaat, een nieuwe catalogus maken. De naam van een catalogusbestand mag uit niet meer dan acht tekens bestaan. dBASE IV gebruikt altijd de extensie (.cat) voor catalogusbestanden.

Als u SET TITLE op ON instelt voor u een nieuwe catalogus maakt, vraagt het systeem u een omschrijving van één regel van het catalogusbestand te geven. In een hoofdcatalogusbestand (Catalog.cat) worden de catalogusnamen opgeslagen samen met de catalogusomschrijvingen die u hebt opgegeven voor iedere catalogus. Wanneer u later het commando SET CATALOG TO ? gebruikt, worden deze omschrijvingen weergegeven zodat u een catalogusnaam kunt selecteren.

Catalogi zijn database-bestanden met een van tevoren bepaalde bestandsstructuur. Wanneer u een catalogusbestand maakt of kiest, wordt dit automatisch geopend in werkgebied 10. Als een catalogus is geopend, is werkgebied 10 gereserveerd voor het bijwerken van het catalogusbestand. U kunt dan geen ander database-bestand gebruiken in dat werkgebied.

Als u een nieuwe catalogus opent of maakt, wordt SET CATALOG automatisch op ON ingesteld. Vanaf dat moment worden alle database-bestanden die u gebruikt (en alle bijbehorende bestanden, zoals index-, query-, indelings-, rapport- en etiketbestanden) en alle bestanden die u maakt, aan de catalogus toegevoegd. Wanneer SET CATALOG op ON is ingesteld, worden alle wijzigingen die worden aangebracht in bestanden die horen bij geopende database-bestanden, geregistreerd in het geopende catalogusbestand. De catalogus wordt automatisch bijgewerkt door bepaalde commando's, bijvoorbeeld door CREATE, INDEX en CREATE/MODIFY REPORT of CREATE/MODIFY SCREEN.

Wanneer u een catalogus opent, wordt de inhoud van de catalogus vergeleken met die van de schijf. Als u bestanden hebt verwijderd toen SET CATALOG op OFF was ingesteld, wordt de catalogus bijgewerkt door de bestanden te verwijderen die nog wel in de catalogus maar niet meer op de schijf.

SET CATALOG

Als u een catalogus wilt sluiten, moet u het commando SET CATALOG TO ↵ gebruiken. Als u geen bestanden meer wilt toevoegen aan een geopende catalogus, moet u SET CATALOG instellen op OFF. Als u daarna weer bestanden aan de catalogus wilt toevoegen, moet u SET CATALOG weer op ON instellen.

Met SET CATALOG OFF wordt, in tegenstelling tot SET CATALOG TO ↵, het catalogusbestand niet gesloten. Wanneer SET CATALOG op OFF is ingesteld, kunt u MODIFY REPORT ? gebruiken. Er wordt dan een menulijst geactiveerd met de .frm-bestanden in de geopende catalogus. Omdat de bestanden in de catalogus zijn verbonden met het database-bestand waarvoor ze zijn gebruikt of gemaakt, worden alleen de bestanden weergegeven die relevant zijn voor het actieve database-bestand. Als echter het commando SET CATALOG ↵ wordt gebruikt, levert MODIFY REPORT ? een lijst op met alle .frm-bestanden in de actieve bestandsindex.

Catalogusstructuur

Alle catalogi hebben dezelfde bestandsstructuur. De structuur is te zien in tabel 3-1.

Tabel 3-1 Catalogusstructuur

Num	Veldnaam	Veldttype	Breedte	Dec
1	Path	Teken	70	
2	File_name	Teken	12	
3	Alias	Teken	8	
4	Type	Teken	3	
5	Title	Teken	80	
6	Code	Numeriek	3	
7	Tag	Teken	4	
** Totaal **			181	

Met uitzondering van de velden Title en Tag, worden alle velden automatisch gevuld als een nieuw bestand aan de catalogus wordt toegevoegd.

Path

Dit is de padnaam die wordt gebruikt als de catalogus niet in het standaard station en in de standaard bestandsindex staat.

File_name

Dit is de naam van het bestand, met extensie.

Alias

Als u geen aliasnaam opgeeft, gebruikt dBASE IV de naam van het database-bestand als aliasnaam. Dit veld blijft leeg bij bestanden van een ander type.

Type

In dit veld staat de standaard extensie die het type bestand aangeeft. Zelfs als u een andere extensie hebt opgegegeven bij het maken van het bestand, wordt de standaard extensie ingevoerd in het veld Type. De opgegeven extensie wordt opgenomen in het veld File_name.

Title

Dit veld is optioneel. Als SET TITLE op ON is ingesteld, vraagt dBASE IV om een omschrijving als u een catalogus maakt. De omschrijving mag uit niet meer dan 80 tekens bestaan. Als u de omschrijving wilt wijzigen en de catalogus niet met USE wordt gebruikt, typt u:

```
. USE <catalogusnaam.cat>
. EDIT
```

Als de catalogus wel wordt gebruikt:

```
. SELECT 10
. EDIT
```

Code

Als er een catalogus is geopend, is Code het nummer dat wordt toegewezen aan elk database-bestand dat met USE in gebruik wordt genomen.

Programmabestanden krijgen een 0 toegewezen. Een database-bestand krijgt een nummer wanneer het wordt gemaakt. Elk nieuw database-bestand krijgt een hoger codenummer. Bestanden die bij een ander bestand horen, bijvoorbeeld indexbestanden, indelingsbestanden, query-bestanden en rapportformulierbestanden, krijgen hetzelfde codenummer als het database-bestand waarnaar ze verwijzen.

Tag

Dit veld wordt niet gebruikt.

SET CATALOG

Bestanden toevoegen

Wanneer SET CATALOG op ON is ingesteld, wordt automatisch een nieuw bestand aan de catalogus toegevoegd als u een van de volgende commando's geeft:

COPY STRUCTURE	IMPORT FROM
COPY STRUCTURE EXTENDED	INDEX
CREATE	JOIN
CREATE FROM	SET FILTER TO FILE
CREATE/MODIFY LABEL	SET FORMAT
CREATE/MODIFY QUERY	SET VIEW
CREATE/MODIFY REPORT	SORT
CREATE/MODIFY SCREEN	TOTAL
CREATE/MODIFY VIEW	USE

Als de bestandsnaam al in de catalogus voorkomt, wordt u gevraagd het bestand een titel te geven (als SET TITLE op ON is ingesteld). Met deze commando's kunt u ook bestanden kiezen met de ?-vraag van de actieve catalogus, ongeacht of CATALOG op ON of op OFF is ingesteld.

OPMERKING

Als u SET TITLE op OFF instelt, wordt u niet gevraagd een titel voor een bestand in te voeren. Met het commando SET CATALOG OFF kunt u voorkomen dat de catalogus wordt bijgewerkt.

Zie ook

SET(), SET TITLE

SET CENTURY

Met SET CENTURY kunt u in het jaargedeelte van datums eeuwen invoeren en weergeven.

Syntaxis

SET CENTURY on/OFF

Standaardinstelling

De standaardinstelling van SET CENTURY is OFF. Cijfers die de eeuwen aangeven worden niet weergegeven en kunnen niet worden ingevoerd.

Standaard worden bij datums 2 cijfers gebruikt voor het weergeven van het jaartal (alleen twintigste eeuw). Wanneer de berekening van een datum echter resulteert in een jaartal dat niet in de twintigste eeuw valt, kan die datum in het datumveld worden geplaatst en blijft het jaartal juist. Als het veld wordt bewerkt met een schermgroot commando en SET CENTURY op OFF is ingesteld, wordt het jaar gewijzigd in een datum die in de twintigste eeuw valt.

Als SET CENTURY op ON is ingesteld, wordt het jaar weergegeven met vier cijfers. Hoewel de weergave van de datum dan tien tekens lang is, blijft de veldlengte in de database altijd acht bytes. Dat komt doordat datums intern worden opgeslagen als cijfers. De manier waarop een datum wordt weergegeven, heeft geen invloed op de manier waarop de datum wordt opgeslagen.

Gebruik

Wanneer SET CENTURY op ON is ingesteld, wordt het jaartal van een datum met vier cijfers weergegeven. Wanneer SET CENTURY op OFF is ingesteld, wordt het jaar weergeven met 2 cijfers en wordt aangenomen dat het jaar in de twingste eeuw valt.

De instelling van SET CENTURY heeft gevolgen voor alle schermgrote bewerkcommando's en de weergave van alle datums. Wanneer SET CENTURY op OFF is ingesteld kunnen alleen datums in de twintigste eeuw worden ingevoerd.

Voorbeeld

```
. ? DATE()
14-02-88
. SET CENTURY ON
. ? DATE()
14-02-1988
. SET CENTURY OFF
```

Zie ook

CTOD(), DATE(), DTOC(), DTOS(), SET(), SET DATE, YEAR()

SET CLOCK

Met SET CLOCK kunt u de weergave en beeldschermpositie van de systeemklok bepalen.

Syntaxis

SET CLOCK on/OFF

SET CLOCK TO [<rij>,<kolom>]

Standaardinstelling

De standaardinstelling voor de klok is OFF. Als de klok wordt ingeschakeld, wordt deze standaard weergegeven op rij 0, in kolom 69.

Gebruik

Met het commando SET CLOCK kunt u de systeemklok weergeven en de plaats van de klok instellen. Met SET CLOCK TO kunt u de klok inschakelen en bepalen waar de klok moet worden weergeven.

Wanneer u geen parameters opgeeft bij SET CLOCK TO, worden de coördinaten van de klok weer op 0, 69 ingesteld.

De klok wordt niet onderdrukt door schermgrote commando's, ongeacht de plaats waarop de klok wordt weergegeven.

Zie ook

SET ()

SET COLOR

Met SET COLOR kunt u kleur- en weergave-attributen voor monochrome beeldschermen en kleurenbeeldschermen opgeven.

Syntaxis

SET COLOR ON/OFF

SET COLOR TO [<normaal>],[<speciaal>],[<beeldrand>],[<achtergrond>]

Normaal,speciaal,beeldrand en *achtergrond* zijn aanduidingen van gebieden van het scherm waarvoor verschillende attributen kunnen worden opgegeven. Deze attributen worden verderop in dit gedeelte beschreven.

SET COLOR OF
    NORMAL/MESSAGES/TITLES/BOX/HIGHLIGHT/INFORMATION/
    FIELDS TO [<attribuut>]

Een attribuut is in dit geval een teken of een combinatie van tekens waarmee kleur, helderheid, knipperen, onderstreepte en omgekeerde weergave worden bepaald. Op bepaalde systemen kunnen attributen voor zowel voor- als achtergrond worden opgegeven, maar deze moeten dan wel door een schuine streep (/) van elkaar worden gescheiden. De tekens die als attribuut kunnen worden gebruikt, zijn:

W - wit	G - groen
N - zwart	B - blauw
U - onderstreept	X - geen
I - omgekeerde weergave	+ - helder
R - rood	* - knipperen

De attributen worden verderop in dit gedeelte uitvoeriger behandeld.

Gebruik

Het commando SET COLOR kent drie varianten die hierna per variant zullen worden behandeld.

SET COLOR ON/OFF

Met het commando SET COLOR ON/OFF schakelt u tussen monochrome beeldschermen en kleurenbeeldschermen, vooropgesteld dat het systeem met beide is uitgerust. De standaardinstelling is die welke bij het starten van dBASE IV door het systeem wordt gebruikt. Met de opdrachtregel COLOR = ON of COLOR = OFF uit het bestand Config.db kunt u de instelling wijzigen.

De functie ISCOLOR() wordt gebruikt om vast te stellen of de actieve weergavemodus kleuren ondersteunt. Wanneer ISCOLOR() waar (.T.) oplevert, is het systeem doorgaans uitgerust met een kleurenkaart en dus in staat kleuren weer te geven.

SET COLOR

SET COLOR TO [<normaal>],[<speciaal>],[<beeldrand>],[<achtergrond>]

Telkens wanneer u dBASE IV start, wordt de weergave bepaald door de kleurinstelling in het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV* voor informatie over het instellen van kleuren in Config.db.

Met het commando SET COLOR TO ↵, zonder attributen, wordt de instelling gewijzigd in zwart-wit. Met dit commando herstelt u niet de instelling zoals die in Config.db is opgenomen.

Op de plaats van de parameters <normaal>, <speciaal>, <beeldrand> en <achtergrond> in de al beschreven syntaxis geeft u het attribuut op dat de kleuren aanduidt waarin de betreffende gebieden (indien aanwezig) moeten worden weergegeven.

Met *normaal* en *speciaal* worden gebieden van het scherm bedoeld die op elk type beeldscherm zijn in te stellen. Deze gebieden worden in tabel 3-4 nauwkeuriger gedefinieerd. Tot het normale gebied behoren de weergavegroepen NORMAL (normaal), MESSAGES (meldingen) en TITLES (titels), de meeste elementaire weergavegroepen van dBASE IV. Speciale gebieden, waartoe de weergavegroepen BOX (kaders), HIGHLIGHT (markering), INFORMATION (informatie) en FIELDS (velden) behoren, krijgen op het scherm extra nadruk. Door bijvoorbeeld in te voeren:

```
. SET COLOR TO B, G
```

wordt tekst in de normale gebieden blauw weergegeven en tekst in de speciale gebieden groen. Om de tekst beter zichtbaar te maken kunt u de kleur instellen waartegen de tekst wordt afgebeeld. De kleur van de tekst wordt de voorgrond genoemd, de kleur waartegen de tekst wordt afgebeeld de achtergrond. U kunt doorgaans een attribuut opgeven voor voor- en achtergrond in zowel het normale gebied als het speciale gebied. In dat geval moet u de kleuren voor voor- en achtergrond scheiden met een schuine streep (/). Met het volgende commando wordt in het normale gebied de tekst wit en de achtergrond blauw weergegeven, terwijl in het speciale gebied de tekst zwart en de achtergrond groen wordt weergegeven:

```
. SET COLOR TO W/B, N/G
```

De gebieden *beeldrand* en *achtergrond* zijn niet op alle typen beeldschermen van toepassing. Met de beeldrand wordt het gebied begrensd waarbinnen tekst wordt weergegeven. Bij sommige weergavemodi zoals Enhanced Graphics Adapter (EGA of compatibel), kan de kleur voor de beeldrand niet worden ingesteld. Met het volgende commando worden in het normale gebied de tekst wit en de achtergrond blauw, in het speciale gebied de tekst zwart en de achtergrond groen, en de beeldrand rood weergegeven:

```
. SET COLOR TO W/B, N/G, R
```

U kunt met de derde parameter ook een kleur voor de achtergrond instellen, maar alleen bij systemen die voor het normale en speciale gebied dezelfde achtergrondkleur vereisen. Op zulke systemen stelt u afzonderlijk de voorgrondkleuren in voor zowel het normale als het speciale gebied en vervolgens de kleur waarin in beide gebieden de achtergrond wordt weergegeven. Met het commando:

```
. SET COLOR TO W, N, R, B
```

wordt de gemeenschappelijke achtergrond van het normale en speciale gebied blauw. De tekst wordt in het normale gebied wit, in het speciale gebied zwart weergegeven, terwijl de beeldrand rood wordt weergegeven. Op de meeste systemen wordt het commando voor deze instellingen als volgt ingevoerd:

```
. SET COLOR TO W/B, N/B, R
```

Het is bij de syntaxis voor SET COLOR TO beslist noodzakelijk de volgorde "normaal gebied, speciaal gebied, beeldrand, achtergrond" aan te houden. Als u bijvoorbeeld de instelling van het speciale gebied wilt wijzigen in zwarte voorgrond op groene achtergrond en daarbij de instelling van het normale gebied onveranderd wilt laten, voert u in:

```
. SET COLOR TO , N/G
```

Met de komma geeft u aan dat de instelling voor het normale gebied onveranderd moet blijven. Als u alleen de kleur van de beeldrand in blauw wilt wijzigen, voert u in:

```
. SET COLOR TO ,,B
```

De tekens die worden gebruikt bij de samenstelling van een attribuut, kunnen hoofdletters en kleine letters zijn. Niet alle attributen werken bij elke weergavemodus. U moet dus testen welke combinaties op uw systeem werken. U kunt op een kleurensysteem bijvoorbeeld best een attribuut voor onderstreepte of omgekeerde weergave invoeren, maar deze beide attributen hebben geen effect op de feitelijke weergave op een kleurenbeeldscherm omdat ze bedoeld zijn voor gebruik op een monochroom systeem. Evenzo kan de kleur van de beeldrand niet op alle systemen worden ingesteld.

W (wit), N (zwart), I (omgekeerde weergave) en U (onderstreept) zijn geschikt voor een monochroom beeldscherm.

Behalve W en N kunt u op een kleurensysteem ook RGB-kleuren weergeven. Combinaties van R (rood), G (groen) en B (blauw) leveren verschillende kleuren op. De tekens kunnen in elke volgorde worden ingevoerd. RB, BR, rb en br duiden alle vier de kleur magenta aan. In tabel 3-2 vindt u een overzicht van de kleurenattributen die u met SET COLOR kunt gebruiken.

Tabel 3-2 Kleurattributen

Voorgrond en achtergrond		Alleen voorgrond	
teken	*kleur*	*teken*	*kleur*
W	wit	W+	helder wit
N	zwart	N+	grijs
R	rood	R+	roze
G	groen	G+	licht groen
B	blauw	B+	licht blauw
RG (of GR)	bruin	RG+ (of GR+)	geel
RB (of BR)	magenta	RB+ (od BR+)	helder magenta
GB (of BG)	cyaan	GB+ (of BG+)	licht cyaan
RGB	wit	RGB+	helder wit

Zoals u ziet heeft het gebruik van het teken (+), dat wordt gebruikt voor het instellen van de helderheid, als attribuut alleen effect bij de instelling van de voorgrondkleur. Zelfs wanneer u (+) expliciet opneemt in het attribuut voor de achtergrond, wordt de voorgrond helder afgebeeld in plaats van de achtergrond. Met het volgende commando wordt de voorgrond helder wit en de achtergrond blauw weergegeven:

```
. SET COLOR TO W/B+
```

Door X of N/N als attribuut te gebruiken worden de voorgrond en de achtergrond allebei zwart weergegeven. U kunt die instelling gebruiken als u bepaalde tekens op een bepaald gedeelte van uw scherm onzichtbaar wilt maken. Met het commando:

```
. SET COLOR TO ,X
```

kunt u een wachtwoord dat bij een GET-commando wordt ingevoerd, onzichtbaar maken. Door een asterisk in het attribuut op te nemen wordt de tekst knipperend weergegeven. Net als (+) heeft de asterisk alleen invloed op de voorgrond. Zelfs wanneer u (*) expliciet in het attribuut voor de achtergrond opneemt, wordt alleen de tekst op de voorgrond knipperend weergegeven. Met het commando:

```
. SET COLOR TO W/B+*
```

wordt tekst in helder wit en knipperend afgebeeld tegen een blauwe achtergrond.

Tabel 3-3 Tekens voor attributen

Teken	Opmerking
I	omgekeerde weergave, alleen bij monochrome systemen
U	onderstreept, alleen bij monochrome systemen
X	zwarte voorgrond op zwarte achtergrond, onderdrukt de weergave
+	helderheid, alleen van toepassing op voorgrondkleur
*	knipperen, alleen van toepassing op voorgrondkleur

SET COLOR OF NORMAL/MESSAGES/TITLES/BOX/HIGHLIGHT/INFORMATION/FIELDS TO <kleurattribuut>

In tabel 3-4 ziet u de namen die zijn toegewezen aan een vooraf gedefinieerde weergavegroep. De eerste drie (NORMAL, MESSAGES en TITLES) worden samen het normale gebied genoemd. De laatste vier (BOX, HIGHLIGHT, INFORMATION en FIELDS) behoren tot het speciale gebied. Als u de kleurinstelling van een van beide gebieden verandert, verandert u de kleuren van zowel NORMAL, MESSAGES als TITLES. Met de derde variant van de syntaxis kunt u de kleurinstelling van de weergavegroepen binnen hun gebieden afzonderlijk wijzigen. U kunt bijvoorbeeld de weergave van de groep MESSAGES (zie tabel 3-4) wijzigen zonder dat de weergave van NORMAL of TITLES wordt gewijzigd.

Tabel 3-4 Weergavegroepen

Weergavegroepen in het normale gebied	Weergave van:
NORMAL	@...SAY-uitvoer zonder het sleutelwoord COLOR Niet geselecteerde velden in BROWSE Ontwerpscherm van het opmaakprogramma (voor rapporten, etiketten en formulieren)1 Niet geselecteerde tekst op ontwerpscherm Niet geselecteerde niet gekleurde weergave alleen veld-templates (bij rapporten, etiketten en formulieren) Statisch memo, randen vensters (bij rapporten, etiketten en formulieren) Niet geselecteerde voorwaarden in QBE-bestandsstructuur Niet geselecteerde voorwaarden in QBE-voorwaardekader Uitdrukkingen in rekenvelden in QBE Randen van niet geselecteerde niet gekleurde kaders in opmaakprogramma (voor rapporten, etiketten en formulieren) Niet gekleurde randen van kaders getekend met @...TO

(wordt vervolgd)

	Tekst van de gebruiker in de Applicatiegenerator
	Scoreboard-gebied
	Ontwerpraamwerk database (CREATE/MODIFY STRUCTURE)
	Kolomnamen in database-ontwerp (CREATE/MODIFY STRUCTURE)
	Veldnummers in database-ontwerp (CREATE/MODIFY STRUCTURE)
MESSAGES	Melding op informatieregel[2]
	Foutmeldingen in foutvenster2
	Aangehaalde commandoregel in foutvenster3
	Meldingen stuurregel2
	Beschikbare, niet geselecteerde opties in menu's en lijsten2
	Niet beschikbare opties in menu's en lijsten3
	Inhoud van bestandenvenster2
	Inhoud van hulpvenster
	Vette tekst in hulpvenster2
	Inhoud van keuzevenster
	Niet geselecteerde knoppen van keuzevenster3
	Niet geselecteerde knoppen van foutvenster3
	Niet geselecteerde knoppen van hulpvenster3
TITLES	Titels bij LIST/DISPLAY
	Titels van hulpvensters
	Paneelnamen in Control Center-scherm2
	Catalogustitel en -bestandsnaam in Control Center
	Bestandsnaam en -omschrijving in Control Center
	Titels van veldnamen bij BROWSE
	Tabelkolommen bij BROWSE3
	Naam bestandsindex in een keuzelijst
	Niet geselecteerde zoneregels rapportontwerp
	Veldnamen en bestandsnaam van QBE-bestandsstructuur
	Veldnamen en visienaam van QBE-visiestructuur
	Raamwerk van QBE-visie3
	Raamwerk van QBE-bestandsstructuur3
	Raamwerk van rekenvelden in QBE3
	Rand van QBE-voorwaardevenster3
	Titel van QBE-voorwaardevenster
	Liniaalregel (in de tekstverwerker: MODIFY COMMAND en tekstverwerker memoveld
	Schriftsoorten gedefinieerd met menu **Stijl** in tekstverwerker2
	Einde-rapport-regel in rapportontwerp
	Statische tekst in formulieren van Applicatiegenerator
	Onderstreepte tekst in hulpvenster2

(wordt vervolgd)

(vervolg)

Weergavegroepen in het speciale gebied	Weergave van:
BOX	Randen van menu's Informatie bij bestanden en volgorde **DOS-hulpprogramma's** Randen van bestandenvenster **DOS-hulpprogramma's** en andere lijsten Randen van keuzelijst Randen van keuzevenster Rand van opmaakvenster in etiketontwerp Rand van foutvenster Randen van niet geselecteerde objecten in Applicatiegenerator
HIGHLIGHT	Gemarkeerde opties in menu's en lijsten2 Geselecteerde statische tekst in ontwerpscherm (voor rapporten, etiketten en formulieren) Geselecteerde velden op ontwerpscherm (voor rapporten, etiketten en formulieren) Geselecteerde kader (bij ontwerp rapporten, etiketten en formulieren) Bestandsnaam en omschrijving in Control Center Schaduwcursor in liniaalregel Namen van zones bij rapportontwerp Geselecteerde zones bij rapportontwerp Randen van informatievenster Inhoud van informatievenster Actieve veld in de Applicatiegenerator Statische tekst onder cursor op ontwerpscherm Geselecteerde knop in hulpvenster2
INFORMATION	Klok Randen hulpvenster Statusregel Geselecteerde knop in foutvenster Gemarkeerde knoppen in keuzevenster3
FIELDS	Gegevensinvoergebieden van keuzevenster3 Geselecteerde velden in BROWSE3 Velden in @...GET die mogen worden bewerkt Geselecteerde veld in database-ontwerpscherm Niet geselecteerde velden in de Applicatiegenerator

1 Het attribuut + waarmee de helderheid wordt ingesteld, heeft in dit gebied geen effect. Heldere instelling resulteert in de weergave van de corresponderende donkere kleur. Zo wordt geel (GR+) weergegeven als bruin (GR). Zie tabel 3-2 voor kleuren en helderheid.

2 De kleuren in dit gebied worden altijd helder weergegeven, ongeacht de aanwezigheid van het attribuut +. Zo wordt zwart (N) altijd weergegeven als grijs (N+). Zie tabel 3-2 voor kleuren en helderheid.

3 De kleuren in dit gebied worden helder of donker weergegeven. Omwille van het contrast kunt u beter de heldere kleuren gebruiken, zodat de niet beschikbare opties donker worden weergegeven. Als u de kleurinstelling voor dit gebied helder maakt, worden alle (beschikbare en niet beschikbare) opties helder weergegeven.

Uit tabel 3-4 blijkt dat de kleuren van sommige weergavegebieden niet kunnen worden ingesteld.

Speciale gevallen

De oudere draagbare computers van Compaq met een ingebouwd monochroom beeldscherm, en veel schootcomputers en andere systemen werken alsof ze een kleurenbeeldscherm hebben. De monochrome attributen U en I zorgen dus niet voor onderstreepte en omgekeerde weergave. De instelling van de kleuren wordt vertaald naar groentinten en grijstinten.

Als u het commando DEFINE WINDOW zonder het sleutelwoord COLOR gebruikt, wordt het venster weergegeven volgens de actieve kleurinstelling. Zonder het sleutelwoord COLOR heeft de rand van het venster de actieve kleur van de weergavegroep BOX. U kunt met het sleutelwoord COLOR ook een geheel nieuwe kleur instellen. U kunt de kleurinstelling van een venster expliciet instellen met het sleutelwoord COLOR bij het commando DEFINE WINDOW, maar ook met het commando SET COLOR. In beide gevallen maakt de kleurinstelling deel uit van de definitie van het opgegeven venster. Het venster houdt dan die kleurinstelling, ook als u later de schermkleuren wijzigt en een venster oproept met het commando ACTIVATE WINDOW.

Voorbeelden

Op de volgende manier kunt u voor het normale gebied de tekst in helder wit tegen een blauwe achtergrond weergeven en voor het speciale gebied de tekst in geel tegen een zwarte achtergrond weergeven:

```
. SET COLOR TO W+/B,GR+/N
```

De statusregel maakt deel uit van de groep INFORMATION. U kunt de statusregel andere kleuren geven, bijvoorbeeld helder wit op rood. Voer in:

```
. SET COLOR OF INFORMATION TO W+/R
```

In het normale gebied wordt tekst nog steeds helder wit en de achtergrond blauw weergegeven, terwijl in het speciale gebied de tekst nog steeds geel en de achtergrond zwart wordt weergegeven. De klok gebruikt dezelfde kleuren als de statusregel aangezien beide tot de groep INFORMATION behoren.

Zie ook

@, @... FILL, DEFINE WINDOW, ISCOLOR(), SET(), SET, SET DISPLAY

SET CONFIRM

Met SET CONFIRM kunt u bepalen hoe de cursor wordt verplaatst als het einde van een veld is bereikt.

Syntaxis

SET CONFIRM on/OFF

Standaardinstelling

De standaardinstelling van SET CONFIRM is OFF.

Gebruik

Als SET CONFIRM op OFF is ingesteld, gaat de cursor van het einde van het eerste invoerveld door naar het begin van het volgende veld.

Als SET CONFIRM op ON is ingesteld, blijft de cursor aan het einde van het invoerveld staan totdat op ↵ wordt gedrukt.

Zie ook

SET ()

SET CONSOLE

Met SET CONSOLE kunt u de schermweergave vanuit een programma in- en uitschakelen.

Syntaxis

SET CONSOLE ON/off

Standaardinstelling

De standaardinstelling van SET CONSOLE is ON.

Gebruik

U kunt SET CONSOLE alleen in een programma gebruiken om de uitvoer naar het scherm te onderdrukken. Met SET CONSOLE OFF kunt u voorkomen dat rapporten en programma's die naar de printer worden gezonden, ook op het beeldscherm worden weergegeven.

U kunt SET CONSOLE OFF niet vanaf de commandostip gebruiken.

Wanneer SET CONSOLE op OFF is ingesteld, kunnen nog wel gegevens worden ingevoerd via het toetsenbord. U kunt gegevens typen voor commando's als WAIT en ACCEPT. Het scherm blijft echter leeg. U kunt de aanwijzingen van deze commando's en de gegevens die u typt, dus niet zien.

@...SAY...GET heeft een hogere prioriteit dan de instelling van CONSOLE en wordt weergegeven ongeacht de status van SET CONSOLE.

Wanneer SET CONSOLE op OFF is ingesteld, worden foutmeldingen en aanwijzingen niet onderdrukt.

Zie ook

SET()

SET CURRENCY

Met het commando SET CURRENCY kunt u bepalen welk symbool moet worden gebruikt als valutateken.

Syntaxis

SET CURRENCY TO [<Tuitdr>]

Gebruik

Wanneer u het commando CURRENCY samen met FUNCTION- of PICTURE-clausules gebruikt, wordt het valutateken dat wordt weergegeven, gewijzigd. Met dBASE IV kunt u maximaal negen tekens gebruiken voor dit symbool.

Voorbeelden

```
. SET CURRENCY TO "f"
. STORE 123456,78 TO Getal
. @ 10,0 SAY GETAL PICTURE "@f"
```

Het resultaat:

```
f123456,78
```

Er volgt nu een voorbeeld van het gebruik van de template *f*. U ziet dat u geen scheidingsteken gebruikt bij het invoeren van een getal.

```
. SET POINT TO ","
. SET SEPARATOR TO "."
. SET CURRENCY TO "f"
. STORE 123456,78 TO Getal
. @ 10,0 SAY Getal PICTURE "fff.fff.fff,99"
```

Het resultaat:

```
fffff123.456,78
```

Om te voorkomen dat het teken *f* wordt herhaald, moet u een spatie gebruiken als eerste teken van het valutateken en daarna het commando herhalen. Het resultaat:

```
f123.456,78
```

Zie ook

@, SET CURRENCY LEFT/RIGHT, SET POINT, SET SEPARATOR

SET CURRENCY LEFT/RIGHT

Met SET CURRENCY LEFT/RIGHT kunt u de positie van het valutateken wijzigen van links van het bedrag naar rechts van het bedrag.

Syntaxis

SET CURRENCY LEFT/right

Standaardinstelling

De standaardinstelling van SET CURRENCY is LEFT.

Gebruik

Standaard staat het valutateken links van het bedrag. Met dit commando kunt u het symbool rechts plaatsen voor valuta waarbij het valutateken rechts moet staan, bijvoorbeeld de Franse frank. Om de standaardinstelling te wijzigen moet u het bestand Config.db wijzigen, zoals wordt beschreven in *Aan de slag met dBASE IV*.

Voorbeelden

Als u het valutateken rechts van het bedrag wilt weergeven voor Franse franken, typt u:

```
. SET CURRENCY RIGHT
. SET CURRENCY TO "F"
. STORE 5000.25 TO BEDRAG
. @ 10,0 SAY BEDRAG PICTURE "@$"
5000,25F
```

Zie ook

@, SET CURRENCY, SET POINT, SET SEPARATOR

SET CURSOR

Met het commando SET CURSOR kunt u instellen of de cursor wel of niet wordt weergegeven.

Syntaxis

SET CURSOR ON/off

Standaardinstelling

De standaardinstelling van SET CURSOR is ON.

Gebruik

Wanneer SET CURSOR op OFF is ingesteld, is de cursor verborgen. Met het commando SET CURSOR kunt u de cursor weer zichtbaar maken.

Als u het commando RUN geeft, heeft de cursor de status die voor het starten van dBASE IV actief was. Nadat u vanuit een extern programma terugkeert naar dBASE IV, wordt de cursor weer onzichtbaar als SET CURSOR op OFF is ingesteld

Zie ook

RUN, RUN(), SET(), SET CONSOLE

SET DATE

Met SET DATE kunt u opgeven hoe datums moeten worden weergegeven.

Syntaxis

SET DATE [TO] american/ansi/british/french/german/ITALIAN/japan/usa/mdy/dmy/ymd

Standaardinstelling

De standaardinstelling van SET DATE is ITALIAN.

Gebruik

Dit commando biedt u flexibiliteit bij de invoer en uitvoer van datums. Met dit commando kunt u snel de indeling van datums wijzigen.

U kunt ook de standaardindeling van datums wijzigen door deze optie op te nemen in het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV* voor meer informatie over het wijzigen van het bestand Config.db.

De verschillende indelingen die kunnen worden gebruikt in het commando SET DATE zijn:

AMERICAN	mm/dd/jj
ANSI	jj.mm.dd
BRITISH/FRENCH	dd/mm/jj
GERMAN	dd.mm.jj
ITALIAN	dd-mm-jj
JAPAN	jj/mm/dd
USA	mm-dd-jj
MDY	mm/dd/jj
DMY	dd/mm/jj
YMD	jj/mm/dd

Voorbeelden

```
. Datum = CTOD({"05-11-87"})
05-11-87
. SET DATE ANSI
. ? Datum
87.11.05
. SET DATE BRITISH
. ? Datum
05/11/87
```

Zie ook

DATE(), DMY(), MDY(), SET CENTURY

SET DBTRAP

SET DBTRAP vormt een beveiligingsmechanisme tegen programmafouten wanneer u de uitvoering van een bepaald programma onderbreekt om een ander programma uit te voeren.

Syntaxis

SET DBTRAP ON/off

Gebruik

Een dBASE IV-commando wordt onderbroken als de uitvoering ervan wordt stopgezet om een ander dBASE IV-commando te kunnen uitvoeren.

Stel dat u de uitvoering van een bepaald commando onderbreekt, de omgeving waarin het commando wordt uitgevoerd wijzigt en daarna de uitvoering van het commando voortzet. Er ontstaat zo een mogelijk niet op te lossen probleem. U bladert bijvoorbeeld met BROWSE door een database-bestand waarin een rekenveld staat dat een UDF aanroept. Deze UDF probeert met het commando USE een ander database-bestand in dezelfde werkomgeving te openen. In dit geval is het onmogelijk om na de onderbreking naar de oorspronkelijke werkomgeving terug te keren, want het eerste database-bestand is gesloten. UDF's en enkele ON-commando's verdienen om die reden bijzondere aandacht. In de regel onderbreken deze de uitvoering van commando's om tussenliggende commando's uit te voeren.

Als DBTRAP op ON is ingesteld, beschikt u in dBASE IV over beveiliging op vier niveaus tegen de wijziging van de werkomgeving die tussenliggende commando's kunnen veroorzaken.

1. Als SET DBTRAP op ON is ingesteld, worden alle tussenliggende commando's geblokkeerd die de structuur van het actieve database-bestand of van het actieve venster zouden wissen of wijzigen. Er verschijnen foutmeldingen indien een ON-commando of een UDF een commando bevat dat een geblokkeerde bewerking probeert uit te voeren. Concreet:

   a: Tussenliggende commando's kunnen niet met CLOSE, PACK, MODIFY STRUCTURE, USE of ZAP het bij de onderbreking actieve database-bestand bewerken.

   b: Tussenliggende commando's kunnen bij CLEAR WINDOW, DEFINE WINDOW, RELEASE WINDOW of RESTORE WINDOW niet de naam van het venster bevatten dat op het moment van onderbreken actief is.

2. Als SET DBTRAP op ON is ingesteld, worden bepaalde aspecten van de actieve omgeving opgeslagen voordat tussenliggende commando's worden uitgevoerd. Na beëindiging van de tussenliggende commando's wordt de oorspronkelijke omgeving hersteld. Bij wijziging van deze aspecten van de omgeving verschijnt er geen foutmelding, maar de wijzigingen zijn dan ook alleen maar van kracht tijdens de uitvoering van de tussenliggende commando's. Na de beëindiging worden de wijzigingen ongedaan gemaakt en de oorspronkelijke waarden weer ingesteld.

   Concreet worden als SET DBTRAP op ON is ingesteld de werkomgeving en (indien aanwezig) het venster opgeslagen die actief waren op het moment van de onderbreking. De oorspronkelijke werkomgeving en het oorspronkelijke venster worden hersteld als de

onderbreking die werd veroorzaakt door een UDF of ON-commando, is beëindigd. Ook als door de tussenliggende commando's met SELECT een nieuw werkgebied is geselecteerd, met DEACTIVATE WINDOW het actieve venster inactief is gemaakt of met DEFINE een nieuw venster is gedefinieerd.

3. Door de instelling van SET DBTRAP op ON is het niet mogelijk BROWSE of EDIT recursief te gebruiken. U gebruikt BROWSE recursief als u bijvoorbeeld met BROWSE een rekenveld bewerkt dat een UDF aanroept waarin opnieuw het commando BROWSE (of EDIT, CHANGE, INSERT, of APPEND) wordt gegeven. De commando's APPEND, BROWSE, EDIT en INSERT zijn nauw verwant. Daardoor kan geen van deze commando's zichzelf of een van de andere commando's aanroepen.

   De hiervoor beschreven beperking voor recursief gebruik van BROWSE geldt als SET DBTRAP op ON is ingesteld voor elk werkgebied. Het commando EDIT in een bepaald werkgebied kan nooit het commando APPEND geven voor een ander werkgebied. Als DBTRAP op OFF is ingesteld, geldt de gegeven beperking voor recursief gebruik van BROWSE alleen voor het actieve werkgebied. Met het commando EDIT kan een commando APPEND worden aangeroepen, mits het bestand waaraan records moeten worden toegevoegd in een ander werkgebied is geopend.

   De commando's LIST en DISPLAY kunnen niet recursief worden gebruikt, ongeacht de instelling van SET DBTRAP. U kunt met dBASE IV geen database-bestand uitlijsten en de bewerking onderbreken om een nieuw commando LIST of DISPLAY te geven. Recursief gebruik van de verwante commando's LIST en DISPLAY is eveneens geblokkeerd.

4. Wanneer SET DBTRAP op ON is ingesteld, kunt u in de indexuitdrukking of de FOR-clausule van het commando INDEX geen UDF gebruiken. De onderbreking van het maken van een index kan situaties opleveren die voor INDEX onoplosbaar zijn.

Als SET DBTRAP op OFF is ingesteld, zijn de vermelde beperkingen opgeheven, met uitzondering van de restricties betreffende het recursieve gebruik van LIST en DISPLAY en van commando's van het type BROWSE binnen één werkgebied.

WAARSCHUWING
Zorg ervoor dat u weet wat de gevolgen zijn als u SET DBTRAP op OFF instelt en controleer of die gevolgen het programma niet beïnvloeden. Controleer eventueel ook of er andere beveiligingsconstructies werkzaam zijn.

Door het gebruik van UDF's en ON-commando's neemt de kans op programmeerfouten toe. Besteed daarom goed aandacht aan de logische programmastructuur.

Zie ook

FUNCTION, ON ERROR/ESCAPE/KEY

SET DEBUG

SET DEBUG is een hulpmiddel bij het opsporen van fouten in een programma. Met dit commando kunt u bepalen of de uitvoer van SET ECHO naar het beeldscherm of naar de printer wordt gezonden.

Syntaxis

SET DEBUG on/OFF

Standaardinstelling

De standaardinstelling van SET DEBUG is OFF.

Gebruik

Wanneer SET DEBUG op ON is ingesteld, wordt de uitvoer van SET ECHO naar de printer gezonden. Hierdoor wordt de uitvoer van het programma naar het scherm niet verstoord door de weergave die normaal gesproken tijdens het opsporen van fouten op het scherm verschijnt.

Wanneer SET DEBUG op OFF is ingesteld, wordt de uitvoer van SET ECHO naar het beeldscherm gezonden.

Wanneer zowel SET DEBUG en SET ECHO op ON zijn ingesteld, worden foutmeldingen en de uitvoer van de commando's SET TALK (bijvoorbeeld het aantal geïndexeerde records), LIST en DISPLAY *niet* naar de printer gezonden, tenzij SET PRINTER op ON is ingesteld.

Zie ook

DEBUG, SET(), SET ECHO, SET PRINTER, SET TALK

SET DECIMALS

Met SET DECIMALS kunt u het aantal decimalen instellen dat dBASE IV weergeeft bij de resultaten van numerieke functies en berekeningen.

Syntaxis

SET DECIMALS TO <Nuitdr>

<Nuitdr> is een numerieke uitdrukking tussen 1 en 18.

Standaardinstelling

De standaardinstelling is twee decimalen.

Gebruik

SET DECIMALS geldt voor deling, vermenigvuldiging en alle mathematische, trigonometrische en financiële berekeningen.

Het maximale aantal decimalen is 18. Er kunnen dus maximaal 20 cijfers worden weergegeven, inclusief het decimaalteken.

Voorbeeld

Dezelfde berekenigen met een andere instelling van SET DECIMALS:

```
. ? 3/4
0.75
. ? SQRT(4.37)
2.09
. SET DECIMALS TO 4
. ? 3/4
0.7500
. ? SQRT(4.37)
2.0905
```

Zie ook

SET()

SET DEFAULT

Met SET DEFAULT kan de gebruiker bepalen op welk station alle bewerkingen moeten worden uitgevoerd en alle bestanden moeten worden opgeslagen.

Syntaxis

SET DEFAULT TO <station> [:]

Standaardinstelling

Het standaardstation is het station van waaruit u dBASE IV hebt gestart.

Als u van station verandert met SET DEFAULT TO, is de standaardbestandsindex de laatst gebruikte bestandsindex van dat station. Wanneer u bijvoorbeeld naar station A gaat vanuit de bestandsindex \DATA van station C, is de standaardbestandsindex \DATA als u terugkeert naar station C.

Als u het commando SET DEFAULT geeft, wordt niet gecontroleerd of het opgegeven station ook werkelijk bestaat of gereed is. Het station waarnaar u terugkeert als u stopt met dBASE IV, wordt ook niet gewijzigd.

Raadpleeg *Aan de slag met dBASE IV* als u het standaardstation wilt wijzigen in het bestand Config.db.

SET DIRECTORY heeft een hogere prioriteit dan SET DEFAULT en wijzigt het standaardstandaardstation in het nieuwe werkstation. SET DIRECTORY heeft een hogere prioriteit dan SET DEFAULT, omdat het zowel het station als de bestandsindex in een keer wijzigt. Het zorgt er bovendien voor dat uw dBASE-station en -bestandsindex overeenkomen met het station en de bestandsindex van uw besturingssysteem.

Voorbeeld

Als het database-bestand Klanten zich in station A bevindt, u dBASE IV hebt gestart vanaf station C, en u het volgende commando typt:

```
. USE Klanten
```

wordt de foutmelding **Bestand bestaat niet** gegeven. Als u het standaardstation wilt wijzigen en u het database-bestand wilt openen, typt u:

```
. SET DEFAULT TO A:
. USE Klanten
```

Zie ook

SET(), SET DIRECTORY, SET PATH

SET DELETED

Met SET DELETED kunt u bepalen of records met een wismarkering wel of niet worden genegeerd door andere commando's van dBASE IV.

Syntaxis

SET DELETED on/OFF

Standaardinstelling

De standaardinstelling van SET DELETED is OFF.

Gebruik

Wanneer SET DELETED op ON is ingesteld, worden de records met een wismarkering door de meeste commando's genegeerd. Records met een wismarkering worden niet weergeven door LIST of LOCATE. Wanneer u echter de recordwijzer op een bepaald record plaatst, bijvoorbeeld met het commando GO, wordt dat record wel weergegeven, ongeacht de wismarkering.

Bij de commando's INDEX en REINDEX worden ook de records met een wismarkering geïndexeerd, ongeacht de status van SET DELETED.

Wanneer SET DELETED op ON is ingesteld, worden geen records teruggehaald door RECALL ALL, omdat de records met een wismarkering worden genegeerd.

Als er met SET RELATION een relatie is vastgelegd tussen twee bestanden, onderdrukt SET DELETED ON de weergave van records met een wismarkering in het kindbestand. Het gerelateerde record in het ouderbestand wordt wel weergegeven, tenzij het ouderrecord ook van een wismarkering is voorzien.

Voorbeeld

Om SET DELETED in werking te laten treden moet u de recordwijzer verwijderen van het record met de wismarkering. Door het invoeren van:

```
. GO 2
. DELETE
. SET DELETED ON
. EDIT
```

kunt u record 2 nog steeds bewerken omdat u de recordwijzer niet hebt verwijderd nadat u SET DELETED op ON hebt ingesteld.

Zie ook

DELETED(), SET(), SET FILTER, SET RELATION

SET DELIMITERS

Met SET DELIMITERS kunt u bepalen hoe de breedte van velden wordt aangegeven in de schermgrote modus.

Syntaxis

SET DELIMITERS on/OFF

SET DELIMITERS TO [<Tuitdr>/DEFAULT/]

Standaardinstelling

De standaardinstelling van SET DELIMITERS is OFF, zodat invoervelden niet worden afgebakend door scheidingstekens. Als SET INTENSITY op ON is ingesteld, worden de invoervelden in omgekeerde weergave getoond.

Er worden dubbele punten (:) gebruikt om een veld af te bakenen, tenzij u een ander scheidingsteken kiest met het commando SET DELIMITERS TO.

Gebruik

Met SET DELIMITERS TO <Tuitdr> kunt u bepalen welke tekens worden gebruikt om het veld af te bakenen. De tekenreeks moet uit één of twee tekens bestaan. Als u één teken opgeeft, geeft dat teken het begin en het einde van het veld aan. Als u twee tekens opgeeft, geeft het eerste teken het begin van het veld aan en het tweede het einde van het veld aan. Als u meer dan twee tekens opgeeft worden alleen de eerste twee gelezen.

Als SET DELIMITERS niet op ON is ingesteld, worden de tekens die zijn opgegeven met SET DELIMITERS TO niet gebruikt.

Voorbeelden

Als u het begin en het einde van een veld wilt aangeven met #:

```
. SET DELIMITERS TO "#"
. SET DELIMITERS ON
```

Als u het begin van elk veld wilt aangeven met [en het einde met]:

```
. SET DELIMITERS TO "[]"
. SET DELIMITERS ON
```

Als u de scheidingstekens weer wilt instellen op dubbele punten, geeft u geen parameters op of typt u het sleutelwoord DEFAULT:

```
. SET DELIMITERS TO DEFAULT
```

Het commando @...GET gebruikt de tekens die zijn opgegeven bij het laatste SET DELIMITERS-commando. Het volgende programma van dBASE IV is daarvan een voorbeeld:

```
SET TALK OFF
SET DELIMITERS TO "[]"
SET DELIMITERS ON
een = "abc"
twee = "123"
CLEAR
@ 10,10 GET een
READ
SET DELIMITERS TO DEFAULT
@ 11,10 GET twee
READ
```

Zie ook

@, APPEND, CHANGE, EDIT, INSERT, READ, SET(), SET INTENSITY

SET DESIGN

Met SET DESIGN kunt u voorkomen dat bepaalde bestanden worden gemaakt of gewijzigd vanaf de commandostip, vanuit het Control Center of in een procedure.

Syntaxis

SET DESIGN ON/off

Gebruik

Met het commando SET DESIGN kan een programmeur voorkomen dat gebruikers naar de ontwerpmodus in het Control Center gaan (dat wil zeggen, bestanden maken en wijzigen vanuit de panelen **Gegevens**, **Query**, **Formulier**, **Rapport**, **Etiket** en **Toepassing**). Op die manier kan een programmeur het menu van het Control Center gebruiken als een interface voor een beperkt toegankelijke toepassing. Als SET DESIGN op OFF is ingesteld, is de toetscombinatie **Shift-F2** inactief gemaakt en is het niet langer mogelijk om naar de commandostip of het QBE-ontwerpprogramma te gaan vanuit de modi APPEND, EDIT en BROWSE.

Vanaf de commandostip of in een procedure kunnen, wanneer SET DESIGN op ON is ingesteld, de commando's CREATE of MODIFY STRUCTURE, CREATE/MODIFY APPLICATION, CREATE/MODIFY LABEL, CREATE/MODIFY QUERY/VIEW, CREATE/MODIFY REPORT, CREATE/MODIFY SCREEN en MODIFY COMMAND/FILE niet worden gebruikt. Evenmin hebt u in dat geval toegang tot het QBE-ontwerpprogramma vanuit BROWSE of EDIT.

U kunt echter nog wel het commando CREATE VIEW FROM ENVIRONMENT gebruiken.

Als SET DESIGN op ON is ingesteld (de standaardinstelling), kunt u alle mogelijke bestanden maken en wijzigen.

Zie ook

CREATE of MODIFY STRUCTURE, CREATE VIEW FROM ENVIRONMENT, CREATE/MODIFY APPLICATION, CREATE/MODIFY LABEL, CREATE/MODIFY QUERY/VIEW, CREATE/MODIFY REPORT, CREATE/MODIFY SCREEN, MODIFY COMMAND/FILE, SET()

SET DEVELOPMENT

Met SET DEVELOPMENT kunt u een controleprogramma activeren dat de datum van gecompileerde objectbestanden (.dbo) vergelijkt met de datum van het bijbehorende bronbestand (.prg), zodat u nooit een verouderde versie van een objectbestand gebruikt.

Syntaxis

SET DEVELOPMENT ON/off

Standaardinstelling

De standaardinstelling van SET DEVELOPMENT is ON.

Gebruik

Wanneer SET DEVELOPMENT op ON is ingesteld, vergelijkt dBASE IV de datum en tijd van het programmabestand (.prg) met die van het gecompileerde objectbestand (.dbo). De tekstverwerker van dBASE IV (MODIFY COMMAND) en de compiler verwijderen het gecompileerde .dbo-bestand wanneer het bijbehorende bestand wordt gewijzigd. Daarna wordt het gewijzigde bronbestand automatisch opnieuw gecompileerd en wordt een nieuw .dbo-bestand gemaakt.

Wanneer SET DEVELOPMENT op OFF is ingesteld, worden de datum en tijd van bronbestanden en objectbestanden niet vergeleken.

Zie ook

COMPILE, DO, MODIFY COMMAND/FILE, SET()

SET DEVICE

Met SET DEVICE kunt u bepalen of de uitvoer van het commando @...SAY naar het scherm, de printer of een bestand wordt gezonden.

Syntaxis

SET DEVICE TO SCREEN/printer/bestand <bestandsnaam>

Standaardinstelling

De standaardinstelling is SET DEVICE TO SCREEN.

Gebruik

Met het commando SET DEVICE TO PRINTER wordt de uitvoer van @...SAY naar de printer gezonden. Het commando @...GET wordt genegeerd. Een @-commando dat lagere coördinaten opgeeft dan eerder gegeven @-commando's, resulteert in een paginadoorvoer.

Op sommige printers wordt de uitvoer van @...SAY pas afgedrukt wanneer de printerbuffer leeg is. U kunt de buffer leegmaken door het commando EJECT te gebruiken of een blanco regel af te drukken.

U kunt de uitvoer van het commando @...SAY opslaan in een bestand door een nieuwe bestandsnaam op te geven achter het sleutelwoord FILE.

Voorbeeld

Wanneer u de uitvoer van @...SAY naar een bestand wilt zenden:

```
. SET DEVICE TO FILE Tekst.txt
. @ 1,1 SAY "Dit staat op de eerste regel."
. @ 5,5 SAY "Dit staat op de vijfde regel."
. SET DEVICE TO SCREEN
. TYPE Tekst.txt
Dit staat op de eerste regel.



Dit staat op de vijfde regel.
```

Zie ook

@, EJECT, SET FORMAT

SET DIRECTORY

Met SET DIRECTORY worden het station en de bestandsindex van het besturingssysteem vanuit dBASE IV ingesteld.

Syntaxis

SET DIRECTORY TO [[<station:>][<pad>]]

Standaardinstelling

Door SET DIRECTORY zonder padnaam te gebruiken worden het station en de bestandsindex die bij het starten van dBASE IV actief waren, opnieuw actief.

Als met SET DIRECTORY het station van het besturingssysteem wordt ingesteld, wordt daarmee ook het standaardstation van dBASE IV ingesteld. Het station van het besturingssysteem wordt echter niet gewijzigd door een nieuwe instelling van SET DEFAULT

Wanneer u dBASE IV met QUIT beëindigt, worden de bestandsindex en het station weer actief van waaruit dBASE IV is gestart.

Gebruik

<station:> en <pad> vormen een padnaam van het besturingssysteem of een tekenuitdrukking die een padnaam oplevert. De uitdrukking kan bestaan uit een verwijzingsteken voor een station (inclusief dubbele punt), de naam van een bestandsindex of een combinatie van beide. Er mogen geen spaties staan tussen het verwijzingsteken voor het station en de naam van de bestandsindex. Ook mag de naam van de bestandsindex niet worden afgesloten met een schuine streep.

Dit commando combineert de mogelijkheden van de commando's RUN CD <pad>, RUN <station:> en SET DEFAULT TO <station:>. SET DIRECTORY verdient echter de voorkeur omdat u daarmee in een stap de standaard bestandsindexen en stations van zowel dBASE IV als van het besturingssysteem kunt wijzigen.

Wanneer alleen een station wordt opgegeven, wordt de bestandsindex actief die op dat station als laatste actief is geweest (of de hoofdbestandsindex als geen andere bestandsindex actief is geweest).

Tijdens het compileren worden alle tekens tot aan de eerste spatie beschouwd als het pad, als [[<station:>][<pad>]] een constante reeks is. Het pad wordt tijdens het compileren niet gevalideerd als een geldige padnaam.

Raadpleeg *Aan de slag met dBASE IV* voor meer informatie over het instellen van het station en de bestandsindex in het bestand Config.db.

SET DIRECTORY

Voorbeelden

Met het volgende commando wijzigt u het station en de bestandsindex in E:\ACCT:

```
. SET DIRECTORY TO E:\ACCT
E:ACCT\
```

Met het volgende commando wijzigt u de bestandsindex in C:\VOORBLD:

```
. m_path = "C:\VOORBLD"
\VOORBLD
. SET DIRECTORY TO &m_path
```

Zie ook

DIR, RUN, RUN(), SET(), SET DEFAULT(), SET PATH

SET DISPLAY

Met SET DISPLAY kunt u kiezen tussen een monochroom beeldscherm en een kleurenbeeldscherm. U kunt ook het aantal regels instellen dat wordt weergegeven als uw grafische kaart zowel 25 als 43 regels kan weergeven.

Syntaxis

SET DISPLAY TO MONO/COLOR/EGA25/EGA43/MONO43

Gebruik

Dit commando werkt alleen als uw apparatuur monochrome weergave en kleurenweergave ondersteunt en er een EGA-kaart of soortgelijke grafische kaart is geïnstalleerd die 43 regels op het scherm kan weergeven. Als u niet over de benodigde apparatuur beschikt, heeft dit commando geen invloed op de weergave en verschijnt er een foutmelding.

Voorbeeld

Wanneer de weergavemodus door een extern programma op de oorspronkelijke waarde zou kunnen worden ingesteld, kunt u de actieve weergavemodus opslaan onder een geheugenvariabele. Bij terugkeer naar dBASE IV wordt dan de instelling van SET DISPLAY opnieuw van kracht.

```
. xmode = SET("DISPLAY")
. RUN <extern programma>
. SET DISPLAY TO &xmode
```

Zie ook

SET, SET(), SET COLOR

SET ECHO

Met SET ECHO kunt u commandoregels van programma's van dBASE IV weergeven op het scherm of de printer terwijl ze worden uitgevoerd (vanaf de commandostip of vanuit een programmabestand).

Syntaxis

SET ECHO on/OFF

Standaardinstelling

De standaardinstelling van SET ECHO is OFF.

Gebruik

Programma-instructies worden normaal gesproken niet weergegeven wanneer een programma wordt uitgevoerd. SET ECHO ON is een van de vier commando's van dBASE IV die samen met DEBUG kunnen worden gebruikt om fouten in een programma op te sporen. De drie andere commando's zijn SET DEBUG, SET STEP en SET TALK.

Wanneer zowel SET ECHO als SET DEBUG op ON zijn ingesteld, wordt de uitvoer van SET ECHO naar de printer gezonden.

Zie ook

DEBUG, SET(), SET DEBUG, SET PRINT, SET STEP, SET TALK

SET ENCRYPTION

Met SET ENCRYPTION kunt u bepalen of een nieuw gemaakt bestand moet worden gecodeerd als PROTECT wordt gebruikt. Zonder PROTECT heeft SET ENCRYPTION geen invloed op de codeerstatus van een bestand.

Syntaxis

SET ENCRYPTION ON/off

Standaardinstelling

De standaardinstelling van SET ENCRYPTION is ON.

Gebruik

Met het commando SET ENCRYPTION kunt u opgeven of gekopieerde bestanden (dat wil zeggen bestanden die worden gemaakt met de commando's COPY, JOIN en TOTAL) moeten worden gecodeerd. In een gecodeerd bestand is de inhoud vervormd om de inhoud van het originele bestand te verbergen. Een gecodeerd bestand kan alleen worden gelezen nadat de codering is ontcijferd of wanneer het bestand in ontcijferde vorm naar een ander bestand is gekopieerd.

Om toegang te krijgen tot een gecodeerd bestand moet u in het aanmeldingsscherm een gebruikersnaam, een wachtwoord en een groepsnaam opgeven. Het toegangsniveau en het toegangsrecht bepalen of u een gecodeerd bestand kunt kopiëren of niet.

Om een bestand te ontcijferen moet u SET ENCRYPTION instellen op OFF nadat u toegang hebt gekregen tot het bestand. Dat is bijvoorbeeld nodig als u het bestand wilt exporteren met EXPORT. Stel SET ENCRYPTION in op OFF als u de opties van het commando COPY TO nodig hebt.

OPMERKING
Coderen werkt alleen met PROTECT. Wanneer u dBASE IV niet via het aanmeldingsscherm oproept, kunt u geen gecodeerde bestanden gebruiken.

Alle gecodeerde bestanden van een toepassing moeten dezelfde groepsnaam hebben.

Gecodeerde bestanden kunnen niet worden samengevoegd met ongecodeerde bestanden. Beide bestanden moeten of gecodeerd of ongecodeerd zijn als u het commando JOIN geeft.

Gecodeerde bestanden moeten eerst in ontcijferde vorm naar een nieuw bestand worden gekopieerd voordat ze kunnen worden gewijzigd met de commando's COPY STRUCTURE EXTENDED of MODIFY STRUCTURE. Nadat u het bestand hebt gekopieerd, kunt u de bestandsstructuur wijzigen. U kunt het nieuwe bestand coderen door er met PROTECT een toegangsniveau aan toe te wijzen.

Bestanden die zijn gemaakt met het commando CREATE kunt u coderen door een toegangsniveau aan de bestanden toe te wijzen met PROTECT.

Zie ook

PROTECT, SET()

SET ESCAPE

Met SET ESCAPE kunt u de uitvoering van een programma van dBASE IV onderbreken door op **Esc** te drukken. U kunt een verschuivend scherm laten stoppen door op **Ctrl**-S te drukken. In dat geval kunt u deze drie toetsen ook registreren met INKEY().

Syntaxis

SET ESCAPE ON/off

Standaardinstelling

De standaardinstelling van SET ESCAPE is ON.

Gebruik

Wanneer SET ESCAPE op ON is ingesteld en u op Esc drukt terwijl een commando of programma van dBASE IV wordt uitgevoerd, wordt de uitvoering onderbroken en wordt een foutvenster weergegeven:

***** AFGEBROKEN *****
Annuleren, Negeren, Onderbreken? (A, N, of O)

Wanneer SET ESCAPE op OFF is ingesteld, wordt de **Esc**-toets buiten werking gesteld en kan een commando niet worden onderbroken. Wanneer SET ESCAPE op OFF is ingesteld, kunt u ook **Ctrl-S** niet meer gebruiken om een verschuivend scherm te laten stoppen.

OPMERKING
Wanneer SET ESCAPE op OFF is ingesteld, kunt u de uitvoering van het programma niet stoppen zonder de computer opnieuw op te starten. Gebruik SET ESCAPE OFF alleen in programma's die uitgebreid zijn getest, zodat het programma niet in een eindeloze lus terecht komt. Wanneer u een programma onderbreekt door de computer opnieuw op te starten, kan dat leiden tot verlies van gegevens.

Zie ook

INKEY(), ON ERROR/ESCAPE/KEY, READKEY(), SET()

SET EXACT

Met SET EXACT kunt u bepalen of tekenreeksen van gelijke lengte moeten zijn wanneer twee tekenreeksen worden vergeleken.

Syntaxis

SET EXACT on/OFF

Standaardinstelling

De standaardinstelling van SET EXACT is OFF.

Gebruik

Als SET EXACT op OFF is ingesteld, worden eerst de twee meest linkse tekens van de tekenreeksen vergeleken en daarna de daaropvolgende tekens tot het einde van de reeks rechts van de relationele operator. Als de vergelijking tot op dat punt klopt, is de uitkomst van de vergelijking waar (.T.).

Als SET EXACT op ON is ingesteld, moeten de tekenreeksen (op de volgspaties na) identiek zijn, voordat de uitkomst van de vergelijking waar (.T.) is.

dBASE IV evalueert constanten tijdens het compileren en gaat tijdens de uitvoering van het programma niet opnieuw over tot evaluatie. Wanneer in een programma de commandoregel

? “abc” = “a”

wordt opgenomen en het programma wordt gecompileerd terwijl SET EXACT op OFF staat ingesteld, zal het waar (.T.) opleveren omdat alleen het eerste teken van de reeks is vergeleken. Ook als u naderhand SET EXACT op OFF instelt, zal de commandoregel

? “abc” = “a”

waar (.T.) opleveren.

SET EXACT

Voorbeelden

Het vergelijken van uitdrukkingen met constanten levert verschillende uitkomsten op, afhankelijk van de instelling SET EXACT. Zo leveren de volgende vergelijkingen de vermelde uitkomsten op:

```
.. SET EXACT ON
. ? "abc" = "a"
.F.
. ? "a" = "abc"
.F.
. ? "abc" = "abc"
.T.
SET EXACT OFF
. ? "abc" = "a"
.T.
. ? "a" = "abc"
.F.
. ? "abc" = "abc"
.T.
```

Zie ook

SET()

SET EXCLUSIVE

Met SET EXCLUSIVE kan een gebruiker een database-bestand openen voor exclusief gebruik in een multi-user systeem, zodat andere gebruikers *geen* toegang hebben tot dat bestand.

Syntaxis

SET EXCLUSIVE on/OFF

Standaardinstelling

De standaardinstelling van SET EXCLUSIVE is OFF.

Gebruik

Wanneer een bestand in exclusief gebruik is, kunnen andere gebruikers dat bestand niet lezen en er ook niet naar schrijven.

SET EXCLUSIVE wordt automatisch op ON ingesteld als u het commando CREATE of SAVE gebruikt.

Zie ook

COPY INDEXES/TAG, INDEX ON, PROTECT, SET(), SET ENCRYPTION, USE

SET FIELDS

Met SET FIELDS kunt u een lijst met velden opgeven uit één of meer bestanden, om daarmee bepaalde bewerkingen uit te voeren. Het bepaalt ook of die lijst de standaardveldenlijst of uitdrukkingenlijst wordt voor commando's waarbij een veldenlijst kan worden opgegeven. Het leesrecht wordt ingesteld en rekenvelden en jokers kunnen worden opgenomen in veldnamen.

Syntaxis

SET FIELDS on/OFF

SET FIELDS T [<veld> [/R] /<code rekenveld>...]

[,<veld>[/R] /<code rekenveld>...]

SET FIELDS TO ALL [LIKE/EXCEPT <structuur>]

Standaardinstelling

De standaardinstelling van SET FIELDS is OFF. SET FIELDS wordt op ON ingesteld als u een veldenlijst opgeeft met SET FIELDS TO. U kunt SET FIELDS ON niet gebruiken voordat u een veldenlijst hebt opgegeven.

Gebruik

Bij veldopties kunt u een lijst met database-velden en rekenvelden opgeven. Met /R kunt u leesrecht instellen, maar alleen voor database-velden.

De naam van een rekenveld kan een willekeurige geldige uitdrukking van dBASE zijn. Bijvoorbeeld:

Bruto_Sal = Salaris * Uren

Bruto_Sal is een rekenveld dat de waarde krijgt van de uitdrukking Salaris * Uren.

In de structuur wordt de joker * voor een willekeurig aantal tekens gebruikt en de joker ? voor één teken. Met ALL LIKE worden alle velden geselecteerd die overeenkomen met de structuur. Met ALL EXCEPT worden alle velden geselecteerd die niet overeenkomen met de structuur.

Wanneer SET FIELDS op OFF is ingesteld (de standaardinstelling), kunnen alle velden in het actieve database-bestand worden gebruikt. Velden in geopende bestanden in andere werkgebieden kunnen alleen worden weergegeven als u de aliasnaam voor de veldnaam zet.

Met SET FIELDS TO kunt u een lijst opgeven met velden die kunnen worden gebruikt in één of meer bestanden. De opgegeven bestandenlijst wordt ook de nieuwe standaarduitdrukkingenlijst of veldenlijst voor commando's van dBASE IV. De veldenlijst die wordt opgegeven bij SET FIELDS TO is niet actief, tenzij SET FIELDS is ingesteld op ON.

Met SET FIELDS TO ↵ kunt u de velden uit de veldenlijst verwijderen. Met SET FIELDS TO ALL kunt u alle velden van het actieve database-bestand opnemen in de veldenlijst.

Wanneer een veldenlijst is ingesteld, kunt u met SET FIELDS TO velden aan de actieve veldenlijst toevoegen. Om velden aan de lijst toe te voegen van bestanden die in een ander werkgebied zijn geopend, moet u het werkgebied van dat bestand kiezen en het commando SET FIELDS TO geven. U kunt ook de alias van dat bestand gebruiken bij de veldnaam.

Wanneer SET FIELDS op ON is ingesteld, worden velden die in de actieve veldenlijst staan bij de commando's LIST STRUCTURE en DISPLAY STRUCTURE aangegeven met een >-teken.

OPMERKING
Met SET FIELDS TO kunt u een lijst opgeven met velden uit meerdere bestanden, maar die bestanden worden niet gerelateerd. Met SET RELATION kunt u bestanden in verschillende werkgebieden met elkaar verbinden. Het commando SET FIELDS TO controleert niet of de bestanden mogelijk aan elkaar gerelateerd zijn. Dat dient u zelf te doen.

Commando's waarop SET FIELDS betrekking heeft

De veldenlijst die u opgeeft met het commando SET FIELDS wordt gebruikt voor alle commando's waarbij de optie veldenlijst in de syntaxis staat. Wanneer SET FIELDS op ON is ingesteld maken de volgende commando's gebruik van de veldenlijst die is opgegeven met SET FIELDS TO:

APPEND
AVERAGE
BROWSE
CALCULATE
CHANGE
CREATE/MODIFY VIEW FROM ENVIRONMENT
COPY STRUCTURE
COPY TO
COPY TO ARRAY
DISPLAY
EDIT
EXPORT
JOIN
LIST
SUM
TOTAL

Ook SET CARRY maakt gebruik van de veldenlijst die met SET FIELDS wordt gemaakt.

Wanneer u een veldenlijst vastlegt en vervolgens het commando COPY STRUCTURE geeft, bevat de nieuwe structuur alleen die velden die in de veldenlijst staan. Verwijder daarom eerst de velden uit de veldlijst met het commando SET FIELDS TO ↵

SET FIELDS

WAARSCHUWING
Om de betrouwbaarheid van gegevens te waarborgen wanneer u visiebestanden of de commando's SET FIELDS of SET RELATION gebruikt, moet u aan één bestand tegelijk records toevoegen en moeten alle velden van het database-bestand toegankelijk zijn. Wanneer u de informatie in de *oudersleutel* wijzigt zonder de bijhorende sleutelinformatie in gerelateerde bestanden te wijzigen, vervalt de relatie tussen die records. Maak geen veldenlijst van meerdere ongerelateerde bestanden, tenzij u daar een gegronde reden voor hebt.

U kunt zelfs indexen gebruiken wanneer er in de indexuitdrukking velden worden gebruikt die niet in de actieve veldenlijst staan.

Bij de commando's LOCATE, SET FILTER en SET RELATION wordt de veldenlijst ook niet gebruikt. Met deze commando's hebt u toegang tot alle velden in alle geopende database-bestanden. Met alle andere commando's en functies hebt u alleen toegang tot de velden die in de veldenlijst van SET FIELDS staan.

Wanneer u meerdere database-bestanden gebruikt, kunt u met een alias onderscheid maken tussen velden met dezelfde naam in verschillende werkgebieden. U kunt bijvoorbeeld A-> en B-> voor de veldnamen zetten om de velden van elkaar te onderscheiden.

Wanneer u geen alias bij de veldnaam opgeeft, gaat dBASE als volgt te werk:

- Als SET FIELDS op OFF is ingesteld, wordt alleen het actieve database-bestand doorzocht.
- Als SET FIELDS op ON is ingesteld, wordt in de veldenlijst gezocht naar de opgegeven veldnaam.

Voorbeelden

In het volgende programma worden de velden Artikel, Stukprijs en Aantal van het database-bestand Magazijn weergegeven. Tevens wordt een rekenveld gemaakt met de naam Totaal, dat het produkt van Stukprijs en Aantal weergeeft.

```
. Use Magazijn
. SET FIELDS TO Artikel, Stukprijs, Aantal, Totaal="ƒ"+STR(Stukprijs*Aantal,8,2)
. GOTO 10
. LIST VOLGEND 5
```

Recordnr.	Artikel	Stukprijs	Aantal	Totaal
10	STOEL, BUREAU	1000,00	1	ƒ 1000,00
11	STOEL, KANTINE	350,00	2	ƒ 700,00
12	BOEKENKAST	125,00	1	ƒ 125,00
13	LAMP, STAAND	150,00	3	ƒ 450,00
14	LAMP, STAAND	165,00	1	ƒ 165,00

In het volgende voorbeeld wordt de optie ALL LIKE opgegeven bij SET FIELDS TO om toegang te krijgen tot alle records in het database-bestand Afnemers waarvan de tekenreeks "naam" voorkomt in de veldnaam.

```
. Use Afnemers
. SET FIELDS TO ALL LIKE *.naam.
. LIST

Recordnr.    ACHTERNAAM    VOORNAAM
        1    Wijsman       Fred
        2    Baljuw        Sandra
        3    Martijn       Rinus
        4    Peters        Kim
        5    Yzinga        Ger
        6    Tiemese       Gemma
        7    Vermeer       Jaco
        8    Beckman       Riener
```

Zie ook

CLEAR FIELDS, CREATE/MODIFY VIEW, CREATE VIEW FROM ENVIRONMENT, SET(), SET CARRY, SET RELATION, SET SKIP, SET VIEW

SET FILTER

Met SET FILTER kunt u instellen dat alleen de records van een database-bestand worden weergegeven die voldoen aan de opgegeven voorwaarde.

Syntaxis

SET FILTER TO [FILE <bestandsnaam>/?] [<voorwaarde>]

Standaardinstelling

Met SET FILTER TO ↵ wordt het filter voor het actieve database-bestand uitgeschakeld.

Met SET FILTER TO FILE voegt u een filterbestand (query-bestand) toe aan de catalogus, indien er een catalogus is geopend en SET CATALOG op ON is ingesteld.

Gebruik

SET FILTER heeft alleen invloed op het geopende database-bestand in het werkgebied waarin het commando wordt gegeven. U kunt dus een aparte filtervoorwaarde opgeven voor elk geopend database-bestand.

De voorwaarden van SET FILTER gelden voor alle commando's waarvoor een database-bestand in gebruik moet zijn, zoals AVERAGE, BROWSE, EDIT en REPORT.

Met SET FILTER TO FILE <bestandsnaam> wordt de filtervoorwaarde gelezen die is opgeslagen in een bestand dat is gemaakt met CREATE/MODIFY QUERY van dBASE III PLUS. U kunt met dit commando geen dBASE IV-query-bestand (.qbe) gebruiken. In dBASE IV kunt u wel query-bestanden van dBASE III PLUS gebruiken, maar u kunt die bestanden niet wijzigen.

Met SET FILTER TO <voorwaarde> kunt u een filter instellen aan de hand van een geldige uitdrukking van dBASE IV. De filtervoorwaarde kan ieder geldig gegevenstype zijn, bijvoorbeeld een tekenveld (SET FILTER TO Achternaam = "Janssen") of een datumveld (SET FILTER TO Vertrek = {01-01-91}). Beide voorwaarden kunnen ook worden gecombineerd, bijvoorbeeld SET FILTER TO "Achternaam = Janssen .AND Vertrek {01-01-91}.

Filters worden pas actief nadat de recordwijzer in het bestand is verplaatst. De beste manier om een filter te activeren is het commando GO TOP of SKIP te geven zodat de recordwijzer wordt verplaatst.

Voorbeelden

Op de volgende manier kunt u een filter instellen voor alle records waarvan het datumveld Dat_trans gelijk is aan 20-03-87 of later:

```
. Use Transact
. SET FILTER TO Dat_trans >={20-03-87}
. LIST

Recordnr.  Klant_nr   Bestel_nr   Dat_trans   Factuur   Totaal_rek
        8  L00001     87-112      20-03-87    .T.          700.00
        9  A00005     87-113      24-03-87    .T.          125.00
       10  B12000     87-114      30-03-87    .F.          450.00
       11  C00001     87-115      01-04-87    .F.          165.00
       12  A10025     87-116      10-04-87    .F.         1500.00
```

Wanneer u opdracht geeft naar een bepaald record te gaan, wordt de voorwaarde van SET FILTER TO genegeerd. Om verder te gaan met het vorige voorbeeld:

```
. GOTO 3
Transact: Recordnr. 3
. DISPLAY

Recordnr.  Klant_nr   Bestel_nr   Dat_trans   Factuur   Totaal_rek
        3  C00002     87-107      12-02-87    .T.         1250.00
```

Zie ook

CREATE/MODIFY QUERY, SET(), SET DELETED

SET FORMAT

Met SET FORMAT kunt u een formulier kiezen dat wordt gebruikt door de commando's READ, EDIT, APPEND, INSERT, CHANGE en BROWSE.

Syntaxis

SET FORMAT TO [<naam indelingbestand>/?]

Gebruik

De indelingsbestanden kunnen worden gemaakt met CREATE/MODIFY SCREEN. Als u CREATE/MODIFY SCREEN gebruikt, wordt eerst een schermbestand (.scr) gemaakt. Kies de optie **Formulier opslaan** uit het menu **Opmaak** om een .fmt-bestand maken. Dit indelingsbestand kan worden geactiveerd met SET FORMAT TO wanneer u records bewerkt of wijzigt in het bijbehorende database-bestand.

Met het commando SET FORMAT TO activeert u het opgegeven indelingsbestand zodat het wordt gebruikt bij schermgrote bewerkcommando's.

Wanneer u het commando SET FORMAT TO geeft, wordt eerst gezocht naar een gecompileerd indelingsbestand met de extensie .fmo. Als dat niet wordt gevonden, wordt gezocht naar een indelingsbestand met de extensie .fmt. Wanneer u een .fmt-bestand voor het eerst gebruikt, wordt er een gecompileerde versie van het bestand gemaakt met de extensie .fmo. Vanaf dat moment gebruikt SET FORMAT TO het .fmo-bestand. Wanneer er geen .fmo-bestand en geen .fmt-bestand wordt gevonden, kan het opgegeven indelingsbestand niet worden gebruikt.

U kunt een indelingsbestand het beste wijzigen met het commando MODIFY SCREEN, omdat dan zowel het .scr-bestand als het .fmt-bestand worden bijgewerkt. U moet het indelingsbestand niet wijzigen met MODIFY COMMAND, omdat daarmee het schermbestand niet wordt gewijzigd, waardoor het indelingsbestand en het schermbestand van elkaar verschillen.

De indelingsbestanden van dBASE III Plus met de extensie .fmt worden met het dBASE IV-commando SET FORMAT gecompileerd tot bestanden met de extensie .fmo.

In een indelingsbestand kunnen drie delen worden onderscheiden. Het eerste deel is de initialisatie. Hierin mogen alle commando's worden gebruikt, behalve @-commando's, READ-commando's of commando's die het indelingsbestand uitschakelen. In dit deel kunt u geheugenvariabelen definiëren die daarna kunnen worden gebruikt in de twee andere delen van het indelingsbestand.

In het tweede deel van het indelingsbestand wordt de opmaak van het formulier bepaald. In dit deel kunnen alleen @-commando's worden gebruikt. De enige uitzondering is het commando READ, waarmee het einde van een pagina kan worden aangegeven.

Het derde deel is de beëindiging. In dit deel kunnen dezelfde commando's worden gebruikt als in het eerste deel. Geheugenvariabelen die zijn gemaakt in het eerste deel van het indelingsbestand, kunnen in dit deel worden opgeheven.

OPMERKING
Gebruik geen READ-commando aan het einde van het .fmt-bestand. dBASE IV interpreteert het READ-commando dan als een pagina-einde en de cursor blijft net zo lang in de hoek van het scherm staan totdat u op een toets drukt.

OPMERKING
Als u in een programma een indelingsbestand opent, moet u een @...GET invoeren, gevolgd door READ. READ opent het indelingsbestand. Zorg ervoor dat het indelingsbestand na gebruik in een programma wordt gesloten zodat andere variabelen kunnen worden gelezen.

Wanneer de nieuwe sleutelwoorden van het commando @, zoals VALID en WHEN, worden gebruikt in een indelingsbestand, is dat bestand niet compatibel met dBASE III PLUS.

Wanneer SET FORMAT niet wordt opgegeven, gebruiken APPEND, CHANGE, EDIT en INSERT het standaard weergave- en invoerformulier van dBASE IV.

De optie FORMAT van het commando BROWSE is van toepassing op alle @...GET-opties in het indelingsbestand bij de bladertabel (behalve op de positie van de velden op het scherm)

Wanneer een catalogus is geopend en SET CATALOG op ON is ingesteld, wordt de catalogus bijgewerkt door het commando SET FORMAT

Wanneer er een catalogus is geopend, kunt u met SET FORMAT TO ? een bestandsindex weergeven met alle .fmt-bestanden van het actieve database-bestand. Wanneer er geen catalogus is geopend, geeft SET FORMAT TO ? een bestandsindex van het actieve station.

In elk werkgebied waarin een database-bestand is geopend, kan een indelingsbestand worden geopend. Met SET FORMAT TO ↵ en CLOSE FORMAT kunt u het geopende indelingsbestand in het actieve werkgebied sluiten.

Zie ook

@, APPEND, BROWSE, CHANGE, CLOSE, EDIT, INSERT, READ, SET(), SET DEVICE

SET FULLPATH

Met het commando SET FULLPATH kunt u instellen of bij de functies MDX(), NDX() en DBF() de volledige padnaam wordt weergegeven of niet.

Syntaxis

SET FULLPATH on/OFF

Standaardinstelling

De standaardinstelling van SET FULLPATH is OFF.

Gebruik

In dBASE IV worden de volledige pad- en bestandsnaam weergegeven bij functies die een bestandsnaam opleveren. In dBASE III PLUS echter worden bij dezelfde functies alleen het station en de bestandsnaam weergegeven.

U kunt het commando SET FULLPATH OFF opnemen in programma's die de functies MDX(), NDX() of DBF() gebruiken, zodat de uitkomst compatibel is met dBASE III PLUS.

U kunt dit commando invoeren vanaf de commandostip. U kunt het ook toevoegen aan het bestand Config.db zodat de weergave van de padnaam altijd wordt onderdrukt.

Zie ook

SET()

De wijzigingen en verbeteringen van dBASE IV worden behandeld in *Uitbreidingen en wijzigingen in dBASE IV*.

SET FUNCTION

Met SET FUNCTION kunt u een functietoets programmeren. Telkens wanneer u dan op die functietoets drukt, worden de geprogrammeerde tekens voor de actieve invoer gebruikt.

Syntaxis

SET FUNCTION <Nuitdr>/<Tuitdr>/<toetslabel> TO <Tuitdr>

Gebruik

U kunt de functietoetsen op drie manieren programmeren:

1. Met het submenu **Toetsen** van het menu **Instellen**
2. Met het commando SET FUNCTION vanaf de commandostip of vanuit een programma.
3. In het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV*.

De standaardinstelling van de functietoetsen is te zien in tabel 3-5.

Tabel 3-5 Standaardinstelling van de functietoetsen

Toets	Waarde	Toets	Waarde
F1	HELP;	F6	DISPLAY STATUS;
F2	ASSIST;	F7	DISPLAY MEMORY;
F3	LIST;	F8	DISPLAY;
F4	DIR;	F9	APPEND;
F5	DISPLAY STRUCTURE;	F10	EDIT;

De functietoets **F1** is gereserveerd voor de functie Hulp en kan niet worden geprogrammeerd.

De functietoetsen **F2** tot en met **F10**, **Shift-F1** tot en met **Shift-F9** en **Ctrl-F1** tot en met **Ctrl-F10** kunnen wel worden geprogrammeerd.

Shift-F10 en alle toetscombinaties met **Alt** worden gebruikt voor macro's en kunnen niet worden geprogrammeerd.

De functietoetsen **F11** en **F12** die voorkomen op een uitgebreid toetsenbord, kunnen niet worden geprogrammeerd in dBASE IV.

U kunt maximaal 254 tekens aan iedere functietoets toewijzen.

De toewijzingen van functietoesen vervallen tijdelijk bij BROWSE en EDIT. Om bij BROWSE en EDIT een tekenreeks weer te geven, moet u het commando ON KEY gebruiken, gevolgd door DO, PLAY MACRO of KEYBOARD.

SET FUNCTION

OPMERKING
In programma's heeft ON KEY een hogere prioriteit dan SET FUNCTION. Wanneer met SET FUNCTION en ON KEY verschillende commando's aan dezelfde toets worden toegewezen, hebben de commando's van ON KEY een hogere prioriteit.

Voorbeelden

Zet een puntkomma aan het einde van de functies en commando's die u toewijst aan functietoetsen, zodat de functie wordt uitgevoerd wanneer u op de toets of toetscombinatie drukt. Gebruik aanhalingtekens om het begin en einde van het commando of de functie aan te geven.

In het volgende voorbeeld wordt het commando SET toegewezen aan functietoets **F10**, zodat het commando wordt uitgevoerd wanneer u op **F10** drukt:

```
. SET FUNCTION F10 TO "SET;"
```

U kunt ook meerdere commando's toewijzen aan bijvoorbeeld functietoets **F9**, zodat de commando's CLEAR, USE Afnemers en LIST STRUCTURE worden uitgevoerd als u op **F9** drukt:

```
. SET FUNCTION F9 TO "CLEAR;USE Afnemers;LIST STRUCTURE;"
```

SET HEADINGS

Met SET HEADINGS kunt u instellen of er titels worden weergegeven bij de commando's AVERAGE, DISPLAY, LIST, SUM en TYPE.

Syntaxis

SET HEADINGS ON/off

Standaardinstelling

De standaardinstelling van SET HEADINGS is ON.

Gebruik

De commando's AVERAGE, DISPLAY, LIST en SUM geven een kolomtitel weer boven ieder veld, iedere geheugenvariabele en iedere uitdrukking, terwijl het commando TYPE de bestandsnaam, de datum en het paginanummer op iedere pagina van de uitvoer weergeeft.

De breedte van de kolom bij de weergave van velden wordt bepaald door de lengte van de titel of de breedte van het veld, al naar gelang welke van de twee het langst is.

Voorbeelden

```
. USE Transact
. DISPLAY

Recordnr.    Klant_nr   Bestel_nr   Dat_trans   Factuur    Totaal_rek
        1    A10025     87-105      03-02-87    .T.          1850.00
        .
        .
        .
. SET HEADINGS OFF
. DISPLAY

        1    A10025     87-105      03-02-87    .T.          1850.00
        .
        .
        .
```

Zie ook

SET()

SET HELP

Met SET HELP kunt u bepalen of in het pop-up venster de dBASE IV hulpaanwijzingen verschijnen wanneer een commando niet goed wordt getypt vanaf de commandostip.

Syntaxis

SET HELP ON/off

Standaardinstelling

De standaardinstelling van SET HELP is ON.

Gebruik

Wanneer SET HELP op ON is ingesteld en u een commando niet goed invoert, verschijnen de opties ANNULEREN, BEWERKEN en HULP in een pop-up venster. U kunt het commando annuleren, de syntaxis van het commando bewerken op de commandoregel of het hulpsysteem gebruiken om de syntaxis van het commando op te zoeken.

Zie ook

HELP, SET()

SET HISTORY

Met SET HISTORY slaat u de uitgevoerde commando's op in een logboekbuffer. Met SET HISTORY TO kunt u bepalen hoeveel commando's in de buffer worden opgeslagen.

Syntaxis

SET HISTORY ON/off

SET HISTORY TO <Nuitdr>

Standaardinstelling

De standaardinstelling van SET HISTORY is ON. De standaardinstelling van SET HISTORY TO is 20. U kunt SET HISTORY TO instellen op een waarde tussen 0 en 16000.

Gebruik

Met SET HISTORY kunt u commando's die zijn gegeven vanaf de commandostip, opnieuw weergeven, bewerken of opnieuw uitvoeren. Commando's die in een programma worden uitgevoerd, worden niet opgeslagen in de logboekbuffer.

Met ↑ en ↓ kunt u commando's die eerder zijn uitgevoerd vanaf de commandostip, opnieuw weergeven. Wanneer u de logboekbuffer doorloopt met ↑, worden de commando's weergegeven in omgekeerd chronologische volgorde, van meest recent naar minst recent. De commando's die in de logboekbuffer staan, worden opgeslagen in het geheugen. Met de commando's LIST HISTORY en DISPLAY HISTORY kunt u alle commando's weergeven die in de buffer staan.

Met SET HISTORY TO kunt u bepalen hoeveel commando's in de logboekbuffer worden opgeslagen. Wanneer dit aantal is bereikt, wordt het minst recente commando uit de buffer verwijderd. Als u de waarde van SET HISTORY TO verlaagt, wordt het aantal opgeslagen commando's verminderd, overeenkomstig de nieuwe waarde. Met SET HISTORY TO 0 wist u alle commando's in de logboekbuffer.

De maximale grootte van de logboekbuffer is afhankelijk van de hoeveelheid geheugen die niet wordt gebruikt. U kunt berekenen hoeveel geheugen de logboekbuffer gebruikt door negen bytes op te tellen bij het aantal bytes van elk commando.

Zie ook

DISPLAY HISTORY, LIST HISTORY, SET()

SET HOURS

Met het commando SET HOURS kunt u de tijdweergave instellen op een 12-uurs of 24-uurs klok.

Syntaxis

SET HOURS TO [12/24]

Standaardinstelling

De standaardinstelling van SET HOURS is 24.

Gebruik

Wanneer u de tijdsweergave wijzigt met het commando SET HOURS TO, verandert de weergave van de klok in alle schermgrote commando's.

Met SET HOURS TO wijzigt u de tijdsweergave voor alle schermen waarop de systeemklok wordt weergegeven, zoals BROWSE en het Control Center.

De enige argumenten die bij dit commando kunnen worden opgegeven zijn 12 en 24. U kunt dit commando ook opnemen in het bestand Config.db om altijd een 12-uurs klok te gebruiken. Raadpleeg *Aan de slag met dBASE IV*

Voorbeelden

Om 8 uur 's avonds verandert de tijdsweergave als volgt:

```
. SET HOURS TO 12
8:00:00 pm
```

```
. SET HOURS TO 24
20:00:00
```

Op de volgende manier kunt u de standaardweergave opnieuw instellen:

```
.SET HOURS TO ↵
```

Zie ook

SET(), SET CLOCK

SET INDEX

Met SET INDEX opent u de opgegeven indexbestanden. U kunt zowel indexbestanden (.ndx) als samengestelde indexbestanden (.mdx) openen. U kunt ook opgeven welke index of label de hoofdindex wordt voor het actieve database-bestand.

Syntaxis

SET INDEX TO [<lijst .ndx- of .mdx-bestanden>/? [ORDER <.ndx-bestand> / [TAG]<naam .mdx-label> [OF <.mdx-bestand>]]]

Standaardinstelling

Als u geen extensie opgeeft, wordt eerst geprobeerd een .mdx-bestand te openen en daarna een .ndx-bestand.

OPMERKING
Waneer u een .ndx-bestand en een .mdx-bestand met dezelfde naam hebt, moet u in de SET INDEX-opdracht de bestandsextensie .ndx opgeven als u wilt dat dit bestand wordt geopend.

SET INDEX TO sluit in eerste instantie alle geopende indexen in het actieve werkgebied, behalve het bijbehorende .mdx-bestand. Daarna worden de opgegeven indexen geopend.

Met SET INDEX TO ↵ sluit u alle geopende .ndx-bestanden en .mdx-bestanden in het actieve werkgebied, behalve het bijbehorend .mdx-bestand. U kunt ook het commando CLOSE INDEX gebruiken.

Gebruik

De index uit de clausule ORDER wordt de hoofdindex. De hoofdindex kan zowel een indexbestand (.ndx) als een labelnaam in een samengesteld indexbestand (.mdx) zijn. De hoofdindex bepaalt hoe de recordwijzer wordt verplaatst in het database-bestand.

Bij de commando's FIND en SEEK wordt de hoofdindex gebruikt om records met elkaar te vergelijken. Alle andere geopende indexen worden bijgewerkt wanneer de gegevens in een sleutel worden gewijzigd. Deze indexen hebben echter geen invloed op de recordwijzer.

Met de clausule OF <.mdx-bestand> kunt u opgeven in welk .mdx-bestand de labelnaam is opgeslagen. U hoeft deze clausule alleen te gebruiken als de labelnaam niet in het bijbehorende .mdx-bestand staat, of dezelfde labelnaam voorkomt in twee of meer geopende .mdx-bestanden.

Wanneer slechts één index is geopend, bepaalt die index de indexvolgorde. De clausule ORDER hoeft dan niet te worden gebruikt.

Met het commando SET ORDER kunt u van hoofdindex veranderen zonder de geopende indexen of .mdx-bestanden te veranderen.

Alle geopende indexen, ook de hoofdindex, worden automatisch bijgewerkt wanneer het bijbehorende database-bestand wordt gewijzigd. Alle labelnamen in geopende .mdx-bestanden

worden ook bijgewerkt.

Wanneer u het commando SET INDEX TO geeft, wordt de recordwijzer op het eerste record van het hoofdindexbestand geplaatst.

Speciale gevallen

Hoewel het database-bestand wordt weergegeven in de volgorde die wordt bepaald door de hoofdindex, wordt de fysieke volgorde van de records in het bestand niet gewijzigd. Als u de feitelijke volgorde van het database-bestand herstelt met SET ORDER TO 0 of SET ORDER TO ↵, worden alle indexen nog steeds bijgewerkt. Als u de feitelijke volgorde herstelt met SET INDEX TO ↵, wordt alleen het bijbehorende .mdx-bestand bijgewerkt. U kunt de andere indexen bijwerken door ze opnieuw te indexeren met het commando REINDEX.

Als u records toevoegt aan een database-bestand terwijl een index actief is, worden die nieuwe records aan het einde van het fysieke database-bestand geplaatst. Wanneer u de velden van een nieuw record hebt ingevoerd, wordt de sleuteluitdrukking opgenomen in de hoofdindex en krijgt het record een plaats in de indexvolgorde. Alle geopende indexen worden bijgewerkt, net als de labels in een geopend .mdx-bestand.

Wanneer een index actief is, werkt het commando INSERT net als het commando APPEND. De nieuwe records worden aan het einde van het fysieke database-bestand geplaatst.

Er kunnen maximaal 10 .mdx-bestanden en 10 .ndx-bestanden plus het bijbehorende .mdx-bestand tegelijkertijd zijn geopend. Wanneer een .mdx-bestand is geopend, zijn alle labels in dat bestand ook geopend.

Voorbeelden

Open het indexbestand Klntnaam voor het database-bestand Afnemers. Typ:

```
. USE Afnemers
. SET INDEX TO Klntnaam
Hoofdindex: KLNTNAAM
```

Om het indexbestand Klntnaam te openen en de indexlabel Klant_nr in het bijbehorende .mdx-bestand als hoofdindex te kiezen, typt u:

```
. SET INDEX TO Klntnaam ORDER Klant_nr
Hoofdindex: Klant_nr
```

Maak een label in nieuw niet-bijbehorend .mdx-bestand en maak deze label vervolgens tot hoofdindex. Typ:

```
. USE Afnemers
. INDEX ON Achtnaam + Voornaam TAG Volnaam OF Afn2
. SET INDEX TO Klntnaam, Afn2 ORDER Volnaam
```

Zie ook

CLOSE, COPY INDEXES, DELETE TAG, INDEX, KEY(), MDX(), NDX(), ORDER(), REINDEX, SET ORDER, TAG(), USE

SET INSTRUCT

Met SET INSTRUCT bepaalt u of bij het genereren van formulieren, rapporten en etiketten wel of niet kaders voor aanwijzingen en dBASE-code worden weergegeven.

Syntaxis

SET INSTRUCT ON/off

Standaardinstelling

De standaardinstelling van SET INSTRUCT is ON.

Gebruik

Wanneer SET INSTRUCT op ON is ingesteld, wordt er een keuzevenster weergegeven met de opties die vanuit het Control Center mogelijk zijn wanneer er een bestand is geselecteerd.

Wanneer SET INSTRUCT op OFF is ingesteld wordt de dBASE-code op het scherm weergegeven zoals gegenereerd tijdens CREATE/MODIFY LABEL, CREATE/MODIFY REPORT en CREATE/MODIFY SCREEN.

Zie ook

SET()

SET INTENSITY

Met SET INTENSITY bepaalt u of bij commando's voor schermgrote bewerkingen gebruik wordt gemaakt van omgekeerde weergave op monochroom beeldschermen en van COLOR OF FIELDS op kleurenbeeldschermen.

Syntaxis

SET INTENSITY ON/off

Gebruik

Wanneer SET INTENSITY op ON is ingesteld, worden op een kleurensysteem de gebieden voor gegevensinvoer bij schermgrote bewerkingen zoals @...GET, EDIT en APPEND, weergegeven volgens de instelling van SET COLOR OF FIELDS. Op een monochroom systeem worden de gebieden voor gegevensinvoer in omgekeerde weergave afgebeeld.

Als SET INTENSITY op OFF is ingesteld, worden de gebieden voor gegevensinvoer weergegeven volgens de instelling van SET COLOR OF NORMAL. Het attribuut van dat commando wordt gebruikt door @...SAY. Op een monochroom scherm worden de gebieden normaal, dat wil zeggen niet in omgekeerde weergave afgebeeld.

Wanneer SET INTESITY op OFF is ingesteld, zijn de gebieden voor gegevensinvoer niet meer te onderscheiden van de rest van het scherm. U kunt met het commando SET DELIMITERS de gebieden voor gegevensinvoer van een rand voorzien.

Zie ook

SET, SET(), SET COLOR, SET DELIMITERS, SET DISPLAY

SET LOCK

Met SET LOCK bepaalt u of bepaalde dBASE IV-commando's in een multi-user systeem automatisch database-bestanden vergrendelen.

Syntaxis

SET LOCK ON/off

Gebruik

Database-bestanden worden automatisch vergrendeld om de betrouwbaarheid van uw gegevens te garanderen. Wanneer u een bestand vergrendelt, hebben andere gebruikers alleen leesrecht. Dat houdt in dat zij de de gegevens wel kunnen bekijken, maar niet kunnen wijzigen. Bepaalde commando's die records lezen en samenvatten, vergrendelen automatisch bestanden als SET LOCK op ON is ingesteld.

Als SET LOCK op ON is ingesteld, heeft het gebruik van dergelijke commando's tot gevolg dat het bestand automatisch wordt vergrendeld, ook al heeft een andere gebruiker alleen maar zijn leesrecht gebruikt. De samenvatting van alle records in het database-bestand wordt niet beïnvloed door toevoegingen of verwijderingen die andere gebruikers aanbrengen. Wanneer SET LOCK op OFF is ingesteld, wordt de automatische vergrendeling opgeheven op het moment dat u een commando geeft dat een samenvatting maakt.

U kunt bijvoorbeeld het commando REPORT FORM gebruiken om een globaal rapport van uw database-bestand te maken. Door REPORT FORM wordt het bestand automatisch vergrendeld zodat het resultaat niet wordt beïnvloed doordat andere gebruikers het bestand bijwerken. Voor uw doel kunt u SET LOCK evenwel net zo goed op OFF instellen voordat u het commando REPORT FORM geeft, zodat u niet verhindert dat andere gebruikers naar het database-bestand schrijven. Aangezien u maar een globaal rapport maakt, ondervindt u weinig hinder van het bijwerken van het bestand door andere gebruikers. Voor het maken van een definitief rapport kunt u SET LOCK weer op ON instellen.

Andere commando's die op dezelfde wijze als REPORT FORM afhankelijk zijn van SET LOCK zijn AVERAGE, CALCULATE, COUNT, LABEL FORM en SUM.

Bepaalde commando's die automatisch een bestand vergrendelen, zoals TOTAL, lezen gegevens uit een bestand en maken tegelijkertijd een tweede bestand. Bij deze commando's wordt door SET LOCK op OFF in te stellen wel de vergrendeling van het (gelezen) bronbestand opgeheven, maar niet die van het doelbestand (waar gegevens naar toe worden geschreven). Een bestand dat nieuw wordt gemaakt, is altijd vergrendeld voor exclusief gebruik door de maker.

Andere commando's die op dezelfde wijze als TOTAL afhankelijk zijn van SET LOCK zijn COPY INDEXES, COPY STRUCTURE, COPY TAG, JOIN en SORT. Bij deze commando's wordt als SET LOCK op OFF is ingesteld, de vergrendeling van het bronbestand opgeheven. De vergrendeling van het doelbestand wordt pas opgeheven nadat de uitvoering van de commando's is voltooid.

Zie ook

FLOCK(), SET()

In *dBASE IV in netwerken* wordt verder ingegaan op vergrendeling van records en bestanden, en op automatisch vergrendelen.

SET MARGIN

Met SET MARGIN kunt u de linker kantlijn instellen voor de printer. De weergave op het scherm wordt niet gewijzigd.

Syntaxis

SET MARGIN TO <Nuitdr>

Standaardinstelling

Standaard is de linker kantlijn ingesteld op nul.

Gebruik

De numerieke uitdrukking is het aantal tekenposities vanaf de linker bladrand. Met SET MARGIN stelt u de kantlijn in voor alle afgedrukte uitvoer van dBASE IV.

De waarde van SET MARGIN is gelijk aan de systeemgeheugenvariabele _ploffset, tenzij _ploffset voor een bepaalde procedure als PRIVATE is gedefinieerd. Als u een _ploffset-variabele als PRIVATE definieert, worden de oorspronkelijke waarden van SET MARGIN of _ploffset hersteld zodra u de procedure beëindigt.

De waarden van SET MARGIN en _ploffset kunnen niet bij elkaar worden opgeteld.

De instellingen SET MARGIN TO 10
    _ploffset = 10

resulteren in een linker kantlijn van tien posities en niet van 20.

U kunt de instelling MARGIN = opnemen in het bestand Config.db zodat een standaard linker kantlijn voor de printer wordt ingesteld. Raadpleeg *Aan de slag met dBASE IV*.

Wanneer _wrap waar (.T.) is, wordt de linker kantlijn opgeteld bij de instelling van SET MARGIN. De linker kantlijn wordt bepaald door de systeemgeheugenvariabele _lmargin. Wanneer _wrap onwaar (.F.) is, worden _lmargin en SET MARGIN niet bij elkaar opgeteld.

SET MARGIN

Voorbeelden

U kunt de linker kantlijn als volgt instellen op 10 spaties rechts van de meest linkse printpositie:

```
. SET MARGIN TO 10
. ? _ploffset
10
```

De instelling van de linker kantlijn kan ook worden gewijzigd met _ploffset:

```
. _ploffset=5
5
```

Omdat de kantlijn voor het laatst is gewijzigd met _ploffset, is dat nu de linker kantlijn. Als u de waarde van de linker kantlijn bekijkt met het commando LIST STATUS, ziet u dat deze is ingesteld op 5. Totdat deze instelling wordt gewijzigd, wordt alle uitvoer 5 spaties rechts van de meest linkse positie afgedrukt.

Zie ook

SET()

SET MARK

Met SET MARK kunt u het teken wijzigen dat de dag, de maand en het jaar scheidt in de datumweergave.

Syntaxis

SET MARK TO [<Tuitdr>]

Gebruik

<Tuitdr> is één teken, *gescheiden door aanhalingstekens*, dat de maand, de dag en het jaar scheidt wanneer een datum wordt weergegeven. De standaardinstelling verschilt per land. In Nederland is het een streepje (-). Op de volgende manier kunt u een ander scheidingsteken kiezen:

```
. SET MARK TO ↵
```

Voorbeeld

```
. SET MARK TO "."
. ? DATE()
22.01.88
```

Zie ook

DATE(), DMY(), MDY(), SET CENTURY, SET DATE

SET MEMOWIDTH

Met SET MEMOWIDTH kunt u de breedte van de uitvoer van memovelden instellen.

Syntaxis

SET MEMOWIDTH TO <Nuitdr>

Standaardinstelling

De standaardbreedte van een memo is 50 tekens. Het minimum is 8 en het maximum is 255. U kunt de standaardbreedte ook wijzigen in het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV*.

Gebruik

Met SET MEMOWIDTH TO kunt u de kolombreedte van memovelden in de uitvoer wijzigen. Wanneer de systeemgeheugenvariabele _wrap op waar (.T.) is ingesteld, wordt de breedte van het memo bepaald door de systeemvariabelen _lmargin en _rmargin.

Met de veldopmaak-functie @V (verticaal vergroten) kunt u het memo weergeven in een verticale kolom wanneer _wrap waar (.T.) is. Wanneer @V nul is, worden memovelden uitgevuld in de breedte van SET MEMOWIDTH. Wanneer @V wordt opgegeven, wordt de waarde van @V opgeteld bij de systeemgeheugenvariabele _pcolno. Op die manier kunt u met de functie @V bepalen hoe ?/??-commando's worden afgedrukt.

Zie ook

?/?? (@V FUNCTION), MEMLINES(), MLINE(), SET()

SET MESSAGE

Met SET MESSAGE kunt u een tekenreeks opgeven die als melding op het scherm wordt weergegeven.

Syntaxis

SET MESSAGE TO [<Tuitdr> [AT <Nuitdr>[,<Nuitdr>]]]

Gebruik

De melding kan maximaal 79 tekens lang zijn.

Hoewel POPUP's, MENU's en meldingen door @...GET-commando's een hogere prioriteit hebben dan een melding die is opgegeven met SET MESSAGE TO, wordt de weergavepositie bepaald door de AT-clausule. De melding wordt gecentreerd weergegeven op de opgegeven regel als u wel de AT-clausule gebruikt maar geen kolompositie opgeeft.

Wanneer SET STATUS op OFF is ingesteld, wordt met het commando SET MESSAGE TO <Tuitdr> AT <Nuitdr>, <Nuitdr> op de opgegeven regel en de optioneel opgegeven kolompositie alleen een melding weergegeven tijdens de schermgrote bewerkcommando's (zoals APPEND, EDIT en READ) en wanneer er menu's actief zijn. Met de eerste <Nuitdr> uit de AT-clausule wordt de regel opgegeven. Deze uitdrukking is verplicht als de AT-clausule wordt gebruikt. Met de tweede <Nuitdr> wordt de kolompositie opgegeven waar de tekst van de melding moet beginnen. De waarde voor de regel ligt tussen 0 en de laatste regel van het scherm. De waarde voor de kolompositie ligt tussen 0 en 79.

Als SET STATUS op ON is ingesteld, wordt de AT-clausule genegeerd. De tekst van uw melding wordt dan gecentreerd weergegeven op de onderste regel van het scherm. Dit is de zogenaamde *informatieregel*. Meldingen op de informatieregel worden altijd weergegeven en niet alleen tijdens de schermgrote commando's

Om een door een gebruiker opgegeven melding te wissen moet u SET MESSAGE TO ↵ invoeren.

Tips

Meldingen kunnen op alle plaatsen van het scherm worden weergegeven, maar niet in vensters. Vermijd dat vensters en andere tekst op het scherm met meldingen worden overschreven.

Stel SET SCOREBOARD op OFF in voordat u een melding op regel 0 weergeeft.

Zie ook

SET SCOREBOARD, SET STATUS

SET NEAR

Met SET NEAR wordt de recordwijzer op het record geplaatst dat direct volgt op de mogelijke lokatie van de gezochte sleutel als bij FIND, SEEK of SEEK() geen exact overeenstemmende sleuteluitdrukking is gevonden.

Syntaxis

SET NEAR on/OFF

Standaardinstelling

De standaardinstelling van SET NEAR is OFF.

Gebruik

Wanneer SET NEAR op ON is ingesteld, wordt de recordwijzer geplaatst op het record dat het meest lijkt op de met FIND, SEEK of SEEK () gezochte maar niet gevonden sleuteluitdrukking. Wanneer SET NEAR op OFF is ingesteld en de sleuteluitdrukking niet wordt gevonden, wordt de recordwijzer aan het einde van het bestand geplaatst.

Wanneer SET DELETED of SET FILTER op ON is ingesteld, wordt daar rekening mee gehouden bij het kiezen van een record dat het meest overeenkomt met de sleuteluitdrukking. Records met een wismarkering worden niet gebruikt, evenals records die niet voldoen aan de voorwaarden van SET FILTER.

Wanneer SET NEAR op ON is ingesteld, geldt:

- FOUND() is waar (.T.) als de sleuteluitdrukking is gevonden.
- FOUND() is onwaar (.F.) als de zoeksleutel niet is gevonden. De zoeksleutel wordt dan op het record geplaatst dat het meest lijkt op de gezochte waarde.
- EOF() is onwaar (.F.) als de sleuteluitdrukking niet is gevonden.

Wanneer SET NEAR op OFF is ingesteld, geldt:

- FOUND() is onwaar (.F.) als de sleuteluitdrukking niet is gevonden.
- EOF() is waar (.T.) als de sleuteluitdrukking niet is gevonden. Deze functie levert echter waar (.T.) op, omdat de recordwijzer aan het einde van het bestand wordt geplaatst.

Voorbeeld

Stel dat u in het database-bestand Klanten een record zoekt waarvan het veld Klant_nr begint met "D", of anders het eerste record na de positie waarop dat record had moeten staan:

```
.USE Afnemers ORDER Klant_nr
Hoofdindex: KLANT_NR
. LIST Klant_nr

Recordnr.        Klant_nr
        1        A00001
        5        A00005
        8        A10025
        7        B12000
        3        C00001
        2        L00001
        4        L00002

. SET NEAR ON
. FIND D
Niet gevonden.
. ? EOF()
.F.
. ? FOUND()
.F.
```

De functie EOF() kan niet langer worden gebruikt om te controleren of de sleuteluitdrukking is gevonden. Wanneer SET NEAR op ON is ingesteld, moet u daarvoor de functie FOUND() gebruiken.

Zie ook

EOF(), FIND, FOUND(), SEEK, SEEK(), SET()

SET ODOMETER

Met SET ODOMETER kunt u opgeven hoe vaak de recordteller wordt bijgewerkt bij commando's die het aantal records weergeven.

Syntaxis

SET ODOMETER TO <Nuitdr>

Standaardinstelling

De standaardinstelling is 1. Het maximum is 200.

Gebruik

Met SET ODOMETER kunt u instellen hoe vaak de recordteller op het scherm wordt bijgewerkt bij commando's als COPY en RECALL.

Met SET TALK OFF kunt u de recordteller van het scherm verwijderen.

Zie ook

SET()

SET ORDER

Met SET ORDER kunt u een geopend indexbestand of label tot hoofdindex maken, of de besturing ontnemen aan alle indexen, zonder .ndx-bestanden of .mdx-bestanden te sluiten.

Syntaxis

SET ORDER TO
SET ORDER TO <Nuitdr>
SET ORDER TO [TAG]
    <bestandsnaam> / <.mdx-label> [OF <.mdx-bestand>]

Gebruik

Met het commando SET ORDER kunt u snel een nieuwe hoofdindex kiezen, omdat de indexbestanden niet worden gesloten en geopend, en de recordwijzer niet wordt verplaatst. Set ORDER TO kent drie modi.

Als u geen argument opgeeft bij SET ORDER TO, wordt het database-bestand weer in de oorspronkelijke volgorde gezet. Het kan worden gebruikt voor zowel .mdx-als .ndx-bestanden.

Met SET ORDER TO <Nuitdr> wordt compatibiliteit met dBASE III PLUS gehandhaafd. Deze modus kan alleen worden gebruikt met .ndx-bestanden zonder gekoppelde .mdx-bestanden. Als het database-bestand over een bijbehorend bestand beschikt, moet u dit commando in de vorm SET ORDER TO TAG gebruiken. De numerieke uitdrukking moet een getal tussen 0 en 10 zijn, afhankelijk van het aantal geopende .ndx-bestanden. Het getal dat u opgeeft, komt overeen met de positie van een bestand in de lijst met indexbestanden die is opgegeven met SET INDEX TO of met het commando USE.

Wanneer zowel .mdx- als .ndx-bestanden zijn geopend, moet u met de derde modus een hoofdindex kiezen. Wanneer beide typen indexbestanden zijn geopend, mag geen numerieke uitdrukking worden gebruikt. U moet dan een bestandsnaam opgeven. De bestandsnaam kan een indexbestand zijn of een indirecte verwijzing naar een bestand die kan worden beschouwd als een bestandsnaam. U kunt de positie van het geopende .ndx-bestand niet in de lijst opgeven.

Als de indexsleutel wordt gewijzigd of records worden toegevoegd of geschoond, worden alle geopende indexbestanden bijgewerkt. Met SET ORDER TO 0 blijven de .ndx-bestanden geopend zodat ze kunnen worden bijgewerkt.

Als de label niet in het bijbehorende .mdx-bestand staat of als dezelfde labelnaam in meerdere .mdx-bestanden voorkomt, kunt u met de optie OF <naam .mdx-bestand> opgeven in welk .mdx-bestand de label staat.

Commando's die niet aan de hand van indexbestanden werken, bijvoorbeeld LOCATE, werken sneller wanneer geen indexbestanden zijn geopend. Geef het commando SET INDEX TO voordat u een commando geeft dat de recordwijzer verplaatst zonder de index te raadplegen.

Voorbeelden

Wanneer het database-bestand Afnemers wordt geopend, wordt de label Klantnaam opgegeven als hoofdindex met de optie ORDER van het commando USE. Daarna wordt in plaats van de label Klantnaam het indexbestand Klntnaam gekozen als hoofdindex:

```
.USE Afnemers INDEX Klntnaam ORDER Klantnaam
Hoofdindex: KLANTNAAM
. SET ORDER TO TAG Klntnaam
Hoofdindex: KLNTNAAM
. SET ORDER TO
Database heeft natuurlijke volgorde
```

U kunt op de volgende manier de label Klantnaam weer tot hoofdindex maken:

```
.SET ORDER TO Klantnaam
Hoofdindex: KLANTNAAM
```

Zie ook

DISPLAY, INDEX, KEY(), MDX(), NDX(), ORDER(), REINDEX, SET INDEX, TAG(), USE

SET PATH

Met het commando SET PATH kunt u het zoekpad opgeven dat wordt gevolgd wanneer bestanden niet worden gevonden in de actieve bestandsindex.

Syntaxis

SET PATH TO [<padlijst>]

Gebruik

Een *padlijst* bestaat uit een aantal bestandsindexen die worden gescheiden door komma's of puntkomma's. De maximale lengte van een bestandsindex is 60 tekens. Met het commando SET PATH TO ↵ wordt het actieve pad gewist.

In dBASE IV wordt het pad dat met het DOS PATH-commando is vastgelegd, niet gebruikt. Evenmin gebruiken DOS en het Control Center het pad dat in dBASE IV met het commando SET PATH is vastgelegd.

De bestandsindex van waaruit u dBASE IV hebt gestart, is de standaardbestandsindex, tenzij u een ander station opgeeft. Dat kan met een commando van het besturingssysteem, met het dBASE IV-commando SET DEFAULT, waarmee u een ander station opgeeft, en met het dBASE IV-commando SET DIRECTORY, waarmee u een ander station en een andere bestandsindex opgeeft.

Het commando SET PATH heeft geen invloed op het commando DIR. Met het commando DIR worden de bestanden in de actieve bestandsindex weergegeven, niet de bestanden in de bestandsindexen van het zoekpad. Als u een andere bestandsindex wilt bekijken met DIR, bijvoorbeeld DBDATA, geeft u ook het pad op bij het commando:

```
. DIR \DBASE\DBDATA\
```

Als u een nieuw bestand wilt maken, maar niet in de actieve bestandsindex, geef dan bij CREATE de volledige padnaam op bij de bestandsnaam of gebruik het commando SET DIRECTORY.

Voorbeeld

Het bestand Program.dbo wordt eerst gezocht in de actieve bestandsindex en op het actieve station. Als het bestand niet wordt gevonden zoekt dBASE IV in de bestandsindex C:\DBASE\DBDATA.

```
. SET PATH TO C:\DBASE\DBDATA
. DO Program
```

Zie ook

DIR, SET DEFAULT, SET(), SET DIRECTORY

SET PAUSE

Met het commando SET PAUSE kunt u de werking van het commando SQL SELECT wijzigen. Het commando wordt onderbroken zodra het scherm vol is.

Syntaxis

SET PAUSE on/OFF

Standaardinstelling

De standaardinstelling van SET PAUSE is OFF.

Gebruik

Wanneer SET PAUSE op ON is ingesteld, werkt het commando SQL SELECT ongeveer als het commando DISPLAY van dBASE IV. Na ieder DISPLAY-commando wordt een regel met gegevens naar het beeldscherm gezonden tot het scherm vol is. Op die manier kunt u meerdere schermen met informatie weergeven, waarbij er wordt gewacht voordat het volgende scherm wordt geladen. U kunt tevens kolomtitels weergeven die meer dan één keer kunnen worden opgeroepen vanuit de dBASE IV-code.

Wanneer SET PAUSE op OFF is ingesteld, werkt het commando SQL SELECT ongeveer als het commando LIST van dBASE IV. Alleen voor het eerste record wordt een kolomtitel weergegeven en de resterende records worden afgedrukt zonder dat wordt gepauseerd tussen de diverse schermen.

Zie ook

SET()

Raadpleeg hoofdstuk 6, *SQL-commando's*, voor meer informatie.

SET POINT

Met SET POINT kunt u een ander decimaalteken kiezen.

Syntaxis

SET POINT TO [<Tuitdr>]

Standaardinstelling

Het standaard decimaalteken is een komma (,).

Gebruik

Met het commando SET POINT kunt u de komma als decimaalteken wijzigen in een punt (.). U kunt ook een ander alfanumeriek teken opgeven. Het teken wordt tussen aanhalingstekens opgegeven. U kunt geen cijfer of spatie als decimaalteken opgeven.

De instelling van SET POINT heeft alleen effect op het beeldscherm en de afgedrukte uitvoer. Om de programma's van dBASE IV ook internationaal overdraagbaar te maken wordt voor interne representaties de punt als decimaalteken gebruikt, ongeacht de instelling van SET POINT. U moet daarom een punt gebruiken als decimaalteken wanneer u een constante invoert.

Met het commando SET POINT TO ↵ kunt u het standaard decimaalteken weer instellen.

Voorbeeld

In het volgende voorbeeld ziet u hoe het decimaalteken in een komma wordt gewijzigd, waarna een constante wordt ingevoerd.

```
. SET POINT TO ","
. x = 99.99
                99,99
. ? x
                99,99
```

Zie ook

SET SEPARATOR

SET PRECISION

Met het commando SET PRECISION kunt u instellen hoeveel cijfers intern worden gebruikt bij berekeningen met getallen van het type N.

Syntaxis

SET PRECISION TO [<Nuitdr>]

Standaardinstelling

De standaardinstelling van SET PRECISION is 16. Het aantal cijfers dat u kunt opgeven bij <Nuitdr> is minimaal 10 en maximaal 20.

Gebruik

Getallen van het type N zijn nauwkeurig tot op 15,9 cijfers. Wanneer u SET PRECISION op een getal kleiner dan 15 instelt, wordt dit gebruikt bij rekenkundige bewerkingen en zijn dus alle cijfers nauwkeurig. Is SET PRECISION ingesteld op een waarde tussen 15 en 20, dan kunnen in een getal van meer dan 15 cijfers alle cijfers boven deze grens onnauwkeurig zijn.

Bij bewerkingen waarbij getallen van het type F en N betrokken zijn, is de uitkomst een getal van het type F. Zulke getallen hebben altijd een nauwkeurigheid van 15,9, ongeacht de instelling van SET PRECISION.

Raadpleeg *Aan de slag met dBASE IV* voor informatie over het wijzigen van de standaard PRECISION-waarde in het bestand Config.db.

Zie ook

FIXED(), SET(), SET DECIMALS

SET PRINTER

Met SET PRINTER ON kunt u alle uitvoer die niet is ingedeeld met het commando @...SAY naar de printer of een bestand zenden.

Met SET PRINTER TO kunt u alle uitvoer naar een aparte printer, netwerkprinter of bestand zenden. Wanneer het commando SET PRINTER in een netwerk wordt gegeven, krijgt de file server opdracht de volgende afdrukopdracht uit te voeren.

Syntaxis

Dit commando kan op zes manieren worden gegeven:

SET PRINTER on/OFF

De standaardinstelling is OFF.

SET PRINTER TO <DOS-poort>

De standaardinstelling is het DOS-afdrukprogramma PRN.

SET PRINTER TO \\ <naam computer> \ <naam printer> = <bestemming>

SET PRINTER TO \\SPOOLER

SET PRINTER TO \\CAPTURE

SET PRINTER TO FILE <bestandsnaam>

Gebruik

Met de eerste vijf commando's kunt u uitvoer naar verschillende afdrukeenheden zenden. Met het zesde commando kunt u afdrukbare bestanden maken.

Beeldschermuitvoer naar de printer zenden

Wanneer SET PRINTER op ON is ingesteld, wordt alle beeldschermuitvoer, ook die van de commando's TALK, LIST en ?,

naar de printer gezonden.

Met het commando SET DEVICE TO PRINTER kunt u de uitvoer van het commando @...SAY naar de printer zenden. De instelling van SET PRINTER ON heeft geen invloed op het commando @...SAY.

SET PRINTER

Afdrukken op een andere printer

Met het commando SET PRINTER TO <DOS-poort> kunt u kiezen naar welke printer de uitvoer wordt gezonden. De standaardinstelling is het afdrukprogramma PRN. De uitvoer kan naar een van de drie parallelle poorten worden gezonden (LPT1, LPT2, LPT3) of naar een van de twee seriële poorten (COM1, COM2). U kunt met het commando SET PRINTER TO NUL een virtuele printer creëren voor het testen van toepassingen. De schrijfbewerkingen worden gesimuleerd, terwijl feitelijk geen gegevens worden weggeschreven. Met het volgende commando wordt de uitvoer naar de printer gezonden die is aangesloten op de tweede parallelle poort:

```
.SET PRINTER TO LPT2
```

Wanneer u probeert toegang te krijgen tot een seriële printer en de melding Printer is niet gereed verschijnt terwijl de printer wel op on-line staat, kunt u met behulp van SET PRINTER TO een COM-poort toewijzen aan een LPT-eenheid. Geef in DOS de instructie MODE LPT3= COM1 en typ vervolgens in dBASE IV:

```
. SET PRINTER TO LPT3
```

Afdrukken in een netwerk

In een netwerk wordt door dBASE IV een melding naar de file server gestuurd zodat het laatste afdrukcommando wordt uitgevoerd. U kunt daarom met het commando SET PRINTER TO <DOS-eenheid> het spool-proces starten en de standaard DOS-eenheid opnieuw instellen. Met deze vorm van het commando wordt de spooler van de printer leeggemaakt en wordt de oorspronkelijke bestemming weer ingesteld. Er kunnen ook andere bestemmingen worden ingesteld. U kunt bijvoorbeeld met het commando:

```
. SET PRINTER TO LPT1
```

de lokale parallelle printer als bestemming voor de uitvoer instellen.

In de IBM PC- en 3Com-netwerken:

```
. SET PRINTER TO \\<naam computer>\<naam printer> = <bestemming>
```

<naam computer> is de door het netwerk toegekende naam van de file server van het netwerk.

<naam printer> is de door het netwerk toegekende naam van de printer van het IBM-netwerk. Er kunnen meerdere printers aanwezig zijn in een netwerk.

Bij <bestemming> kan de gemeenschappelijke printer worden opgegeven met LPT1, LPT2 of LPT3. De gemeenschappelijke printer kan zijn aangesloten op de computer van de gebruiker of op een andere computer. Met het volgende commando wordt de uitvoer via de file server naar een gemeenschappelijke Epson printer gezonden die in de file server wordt aangeduid met LPT1:

```
. SET PRINTER TO \\SERVER\EPSON=LPT1
```

Wanneer u een andere printer of spooler hebt gekozen voor de uitvoer, kunt u de standaardprinter weer instellen door de standaardprinternaam op te geven:

```
.SET PRINTER TO <DOS-standaard>
```

Uitvoer opslaan in een bestand

U kunt de uitvoer ook opslaan in een bestand. Het bestand wordt dan opgemaakt voor een bepaalde printer en kan later naar de printer worden verzonden waarvoor het is opgemaakt. Als het bestand wordt gemaakt met het printerbesturingsprogramma Ascii.pr2, wordt een ASCII-bestand gemaakt. Het bestand kan dan naar iedere lijnprinter worden verzonden of via telecommunicatieverbindingen naar een andere computer worden gestuurd.

U kunt als volgt een printerbestand maken:

```
. SET PRINTER TO FILE <bestandsnaam>
```

Bestanden die zijn opgemaakt voor de printer, krijgen de standaardextensie .prt, tenzij u het printerbesturingsprogramma Ascii.pr2 hebt gekozen. In dat geval krijgt het bestand de standaardextensie .txt.

Kies het printerbesturingsprogramma Ascii.pr2 als u een tekstbestand wilt maken:

```
. _pdriver = Ascii.pr2
```

Als u nu de uitvoer opslaat in een bestand, wordt dit een ASCII-bestand.

Als u een bestand wilt opmaken voor een bepaalde printer, bijvoorbeeld de Hewlett-Packard LaserJet, kunt u als volgt het juiste printerbesturingsprogramma opgeven:

```
. _pdriver = "Hplas100.pr2"
```

Kies nu het bestand dat u wilt afdrukken op de LaserJet en sla de uitvoer op in een bestand. Typ:

```
.SET PRINTER TO FILE Bestand
```

Het nieuwe bestand krijgt de naam Bestand.prt en bevat de printerstuurcodes voor een LaserJet. Dit bestand kan worden verzonden vanaf de DOS-aanwijzing op elke PC waarop een LaserJet is aangesloten.

Stel dat het opgemaakte printerbestand op een diskette in station A staat en dat de LaserJet is aangesloten op de seriële poort COM1. U kunt het bestand dan afdrukken door het volgende commando te typen achter de DOS-aanwijzing:

```
COPY A:Bestand.prt COM1
```

De uitvoer is op de juiste wijze opgemaakt voor de LaserJet. Denk eraan dat u met het commando COPY een bestand kopieert en het afdrukprogramma van DOS dus niet wordt opgeroepen. Dat gebeurt wel als u het bestand afdrukt met het commando PRINT van DOS. Als u dat doet, worden alle stuurcodes in het bestand ook afgedrukt.

SET PRINTER

Het commando SET PRINTER TO FILE blijft actief totdat u met het commando SET PRINTER TO de uitvoer weer naar een randapparaat of een spooler zendt. Als u het standaardrandapparaat weer instelt en u een ander printerbesturingprogramma hebt gekozen voor SET PRINTER TO FILE, moet u ook het printerbesturingsprogramma opnieuw instellen. U kunt dat doen door _pdriver in te stellen op de naam van het gewenste printerbesturingsprogramma.

Voorbeelden

Met de volgende commando's zendt u de uitvoer van het rapport Personel naar de netwerkprinter (parallelle printer nummer 1) die is gekoppeld aan de file server ASERVER:

```
. SET PRINTER TO \\ASERVER\PRINTER=LPT1
. REPORT FORM Personel TO PRINTER
. SET PRINTER TO LPT1
```

Met het eerste commando wordt LPT1 gekozen als netwerkprinter. Met het tweede wordt de uitvoer naar de spooler gezonden. Met het derde commando wordt de uitvoer van de spooler naar de printer gezonden en wordt de netwerkprinter opnieuw ingesteld.

Zie ook

?/??, ???, LABEL FORM, LIST/DISPLAY, PRINTJOB/END PRINTJOB, REPORT FORM, SET(), SET DEBUG, SET DEVICE, _pdriver

SET PROCEDURE

Met SET PROCEDURE kunt u een bepaalde procedure openen.

Syntaxis

SET PROCEDURE TO [<naam procedurebestand>]

Standaardinstelling

Als het opgegeven bestand niet op het standaardstation staat en ook niet in het zoekpad van dBASE IV, geeft u ook de stationsaanduiding op.

Als u geen extensie opgeeft bij de bestandsnaam, wordt uitgegaan van een objectbestand met de extensie .dbo. Als er geen .dbo-bestand kan worden gevonden, zoekt dBASE IV naar een programmabestand (.prg) en wordt van het programma een .dbo-bestand gemaakt.

Met SET PROCEDURE TO <naam procedurebestand> wordt het opgegeven procedurebestand geopend. Met SET PROCEDURE TO ↵ of CLOSE PROCEDURE kunt u het actieve procedurebestand sluiten.

Gebruik

In theorie kan een procedurebestand 963 procedures (routines) bevatten. Dit aantal kan echter worden beperkt door het beschikbare geheugen. Het begin van iedere nieuwe routine wordt aangegeven met het commando PROCEDURE. Er kan slechts één procedurebestand tegelijk zijn geopend.

In elke toepassing kunnen procedurebestanden worden opgenomen, maar er kan slechts één procedurebestand tegelijk actief zijn.

Voorbeelden

In de volgende voorbeelden wordt duidelijk gemaakt hoe procedures kunnen worden gebruikt. In het volgende programma wordt een procedurebestand geopend, waarna twee van de procedures in het bestand worden uitgevoerd:

```
* Setproc.PRG - Voorbeeld van SET PROCEDURE TO

SET PROCEDURE TO Proctest      && procedurebestand wordt geopend.
DO Proc1                       && procedure 1 wordt uitgevoerd.
DO Proc2                       && procedure 2 wordt uitgevoerd.
CLOSE PROCEDURE                && procedurebestand wordt gesloten.
* EOP: Setproc.prg
```

SET PROCEDURE

Het procedurebestand Proctest (dat ook een programmabestand van dBASE is) zou er als volgt uit kunnen zien:

```
* Proctest.prg - Procedures

PROCEDURE Proc1
      ? "Dit is een melding van Proc1 van Proctest.prg."
RETURN
* EOP: Proc1

PROCEDURE Proc2
      ? "Dit is een melding van Proc2 van Proctest.prg."
      DO Proc3
RETURN
* EOP: Proc2

PROCEDURE Proc3
      ? "Dit is een melding van Proc3 van Proctest.prg en"
      ? " werd aangeroepen vanuit Proc2 van Proctest.prg."
RETURN
* EOP: Proc3
* EOP: Proctest.prg
```

Als zowel het programma als het procedurebestand in de standaardbestandsindex of in de bestandsindex van SET PATH TO staan, kunt u de procedure als volgt uitvoeren:

```
.DO Setproc
Dit is een melding van Proc1 van Proctest.prg.
Dit is een melding van Proc2 van Proctest.prg.
Dit is een melding van Proc3 van Proctest.prg en
 aangeroepen vanuit Proc2 van Proctest.prg.
```

Zie ook

COMPILE, DO, PARAMETERS, SET()

SET REFRESH

Met SET REFRESH kunt u instellen hoe vaak in een netwerk de gegevens van het werktstation worden bijgewerkt aan de hand van het database-bestand op de file server.

Syntaxis

SET REFRESH TO <Nuitdr>

Standaardinstelling

SET REFRESH is standaard ingesteld op 0, zodat het scherm niet wordt bijgewerkt.

Gebruik

Het commando SET REFRESH werkt alleen in een lokaal netwerk (LAN) en alleen bij bestanden die met het commando CONVERT zijn bewerkt.

De numerieke uitdrukking is een tijdsinterval dat wordt uitgedrukt in seconden en heeft een bereik van 1 tot en met 3.600 seconden.

Als u SET REFRESH op een waarde groter dan 0 instelt en met het commando BROWSE, EDIT of CHANGE records bewerkt, wordt het scherm van een werkstation periodiek bijgewerkt met wijzigingen die door andere werkstations in het bestand op de server zijn aangebracht.

De instelling van SET REFRESH blijf van kracht totdat u het commando QUIT geeft of totdat u met SET REFRESH een andere waarde instelt.

Bij het commando BROWSE worden alle records in de actieve BROWSE-lijst opnieuw vanaf de server gelezen en daarna opnieuw op het scherm van het werkstation weergegeven. Bij EDIT en CHANGE wordt alleen het actieve record opnieuw gelezen omdat deze commando's maar één record weergeven.

Wanneer bij een werkstation het bestand of een aantal records uit het actieve scherm is vergrendeld, kan het scherm niet worden bijgewerkt. Zodra de vergrendeling is opgeheven, wordt het scherm weer periodiek bijgewerkt. Het scherm wordt niet bijgewerkt gedurende de weergave van een BROWSE- of EDIT-menu.

Speciaal geval

U kunt de waarde van REFRESH ook instellen in de betreffende regel in het bestand Config.db.

SET REFRESH

Voorbeeld

Met het volgende commando worden de gegevens op het werkstation bij BROWSE om de vijf seconden bijgewerkt.

```
. USE Personel
. CONVERT
. SET REFRESH TO 5
. BROWSE
```

Zie ook

BROWSE, CONVERT, EDIT, SET()

SET RELATION

Met SET RELATION kunt u het actieve database-bestand koppelen aan een of meer geopende database-bestanden op basis van een gemeenschappelijke sleuteluitdrukking.

Syntaxis

SET RELATION TO [<uitdr> INTO <alias>
[, <uitdr> INTO <alias>...]]

Standaardinstelling

Met SET RELATION TO ↵ zonder parameters kunt u de relatie in het actieve werkgebied ongedaan maken.

Gebruik

Met SET RELATION kunt u het actieve database-bestand koppelen aan elk database-bestand dat voldoet aan de omschrijving bij INTO <alias>. Het actieve bestand dat de relatie bestuurt, wordt het *ouderbestand* genoemd. Het bestand dat aan het ouderbestand wordt gekoppeld, wordt het *kindbestand* genoemd.

U mag geen circulaire relatie definiëren. U mag bijvoorbeeld niet bestand Test1 koppelen aan bestand Test2 en daarna Test2 koppelen aan Test1.

De uitdrukking voor de relatie kan elke uitdrukking zijn die is afgeleid uit de gegevens van het ouderbestand en die overeenkomt met de sleutel van de index waarmee het kindbestand wordt bestuurd. Als het kindbestand niet is geïndexeerd, moet de uitdrukking voor de relatie numeriek zijn en corresponderen met het nummer van het doelrecord in het kindbestand.

Als de optionele numerieke uitdrukking wordt opgegeven, wordt de actieve database altijd verbonden met het recordnummer dat wordt aangeduid met de numerieke uitdrukking. Meestal wordt de functie RECNO() opgegeven als numerieke uitdrukking. In dat geval wordt record 1 van het ouderbestand gekoppeld aan record 1 van het kindbestand, record 2 van het ouderbestand aan record 2 van het kindbestand enzovoort.

Recordwijzer

Als geen overeenkomstig record in het gekoppelde bestand wordt gevonden, wordt de recordwijzer aan het einde van het bestand geplaatst en resulteert EOF() in waar (.T.).

Het commando SET NEAR heeft geen gevolgen voor de kindbestanden van SET RELATION.

Als er een SET SKIP-lijst actief is, wordt de recordwijzer in elk database-bestand voorwaarts verplaatst, te beginnen met het laatste werkgebied in de relatie en ten slotte in het ouderbestand.

Tips

U kunt een werkomgeving, inclusief relaties, opslaan met CREATE VIEW <.vue-bestand> FROM ENVIROMENT. Met dit commando wordt een nieuw .vue-bestand (dBASE III PLUS) gemaakt. CREATE/ MODIFY QUERY/VIEW is een menugestuurd commando waarmee u bestanden kunt koppelen. U kunt er ook relaties mee opslaan in een .qbe-bestand (dBASE IV).

Voorbeelden

In het volgende voorbeeld wordt het database-bestand Afnemers gerelateerd aan het database-bestand Transact. De twee bestanden worden gekoppeld op basis van het veld Klant_nr:

```
. SELECT 1
. USE Afnemers ORDER Klant_nr
Hoofdindex: KLANT_NR
. SELECT 2
. USE Transact
. SET RELATION TO Klant_nr INTO Afnemers
. GO TOP
. LIST NEXT 5 Klant_nr, Afnemers->Klantnaam, Dat_trans, Totaal_rek

Recordnr.   Klant_nr   Afnemers->Klantnaam   Dat_trans   Totaal_rek
        1   A10025     PUNTER PROMOTIONS     03-02-87       1850.00
        2   C00001     L. G. BLOEM & PARTNERS 10-02-87      1200.00
        3   C00002     TIEMESE & KEEFMAN BV  12-02-87       1250.00
        4   C00001     L. G. BLOEM & PARTNERS 23-02-87      1250.00
        5   L00001     BALJUW & CO.          09-03-87        415.00
```

U kunt een database-bestand ook relateren aan meerdere database-bestanden. Het volgende voorbeeld is een vervolg op het vorige:

```
. SELECT 1
. USE Afnemers ORDER Klant_nr
Hoofdindex: KLANT_NR
. SELECT 2
. USE Transact
. SET RELATION TO Klant_nr INTO Afnemers
. GO TOP
. LIST NEXT 5 Klant_nr, Afnemers->Klantnaam, Dat_trans, Totaal_rek

Recordnr.  Klant_nr  Afnemers->Klantnaam   Dat_trans   Totaal_rek
        1  A10025    PUNTER PROMOTIONS     03-02-87      1850.00
        2  C00001    L. G. BLOEM & PARTNERS 10-02-87     1200.00
        3  C00002    TIEMESE & KEEFMAN BV  12-02-87      1250.00
        4  C00001    L. G. BLOEM & PARTNERS 23-02-87     1250.00
        5  L00001    BALJUW & CO.          09-03-87       415.00

. SELECT 3
. USE Magazijn ORDER Bestel_nr
Hoofdindex: BESTEL_NR
. SELECT 2
. SET RELATION TO Klant_nr INTO Afnemers, Bestel_nr INTO Magazijn
. GO TOP
. LIST NEXT 5 Afnemers->Klantnaam, Dat_trans, Magazijn->Artikel

Recordnr.  Afnemers->Klantnaam   Dat_trans   Magazijn->Artikel
        1  PUNTER PROMOTIONS     03-02-87    SOFA, 6 POTEN
        2  L. G. BLOEM & PARTNERS 10-02-87   SOFA,8 POTEN
        3  TIEMESE & KEEFMAN BV  12-02-87    STOEL,BUREAU
        4  L. G. BLOEM & PARTNERS 23-02-87   STOEL,BUREAU
        5  BALJUW & CO.          09-03-87    TAFEL,BIJZET
```

SET RELATION

U kunt een kettingrelatie opbouwen tussen de database-bestanden Artikel, Bestel en Klanten. Het volgende voorbeeld gaat uit van de werkomgeving die is gemaakt met de vorige twee voorbeelden:

```
. SELECT 1
. USE Afnemers ORDER Klant_nr
Hoofdindex: KLANT_NR
. SELECT 2
. USE Transact
. SET RELATION TO Klant_nr INTO Afnemers
. GO TOP
. LIST NEXT 5 Klant_nr, Afnemers->Klantnaam, Dat_trans, Totaal_rek

Recordnr.   Klant_nr   Afnemers->Klantnaam        Dat_trans   Totaal_rek
        1   A10025     PUNTER PROMOTIONS          03-02-87       1850.00
        2   C00001     L. G. BLOEM & PARTNERS     10-02-87       1200.00
        3   C00002     TIEMESE & KEEFMAN BV       12-02-87       1250.00
        4   C00001     L. G. BLOEM & PARTNERS     23-02-87       1250.00
        5   L00001     BALJUW & CO.               09-03-87        415.00

. SELECT 3
. USE Magazijn ORDER Bestel_nr
Hoofdindex: BESTEL_NR
. SELECT 2
. SET RELATION TO Klant_nr INTO Afnemers, Bestel_nr INTO Magazijn
. GO TOP
. LIST NEXT 5 Afnemers->Klantnaam, Dat_trans, Magazijn->Artikel

Recordnr.   Afnemers->Klantnaam   Dat_trans   Magazijn->Artikel
        1   PUNTER PROMOTIONS     03-02-87    SOFA, 6 POTEN
        2   L. G. BLOEM & PARTNERS 10-02-87   SOFA,8 POTEN
        3   TIEMESE & KEEFMAN BV  12-02-87    STOEL,BUREAU
        4   L. G. BLOEM & PARTNERS 23-02-87   STOEL,BUREAU
        5   BALJUW & CO.          09-03-87    TAFEL,BIJZET

.SET RELATION TO Klant_nr INTO Afnemers
. SET ORDER TO Bestel_nr
Hoofdindex: BESTEL_NR
. SELECT 3
. SET INDEX TO              && Het bestand wordt in natuurlijke volgorde gezet.
. SET RELATION TO Bestel_nr INTO Transact
. GO TOP
. LIST NEXT 5 Artikel, Transact->Dat_trans, Afnemers->Klantnaam

Recordnr.   Artikel       Transact->Dat_trans   Afnemers->Klantnaam
        1   SOFA,6 POTEN  03-02-87              PUNTER PROMOTIONS
        2   SOFA,6 POTEN  03-02-87              PUNTER PROMOTIONS
        3   SOFA,8 POTEN  10-02-87              L. G. BLOEM & PARTNERS
        4   STOEL,BUREAU- 12-02-87              TIEMESE & KEEFMAN BV
        5   STOEL,BUREAU- 23-02-87              L. G. BLOEM & PARTNERS
```

Zie ook

CREATE/MODIFY VIEW, SET(), SET FIELDS, SET FILTER, SET SKIP, SET VIEW

SET REPROCESS

Met SET REPROCESS kunt u instellen hoe vaak in een netwerk wordt geprobeerd een record of bestand te vergrendelen voordat een foutmelding wordt gegeven.

Syntaxis

SET REPROCESS TO <Nuitdr>

Standaardinstelling

De standaardinstelling is nul. Het bereik loopt van -1 tot en met 32.000.

Gebruik

Met het commando SET REPROCESS TO kunt u in een netwerk de uitvoering van commando's wijzigen. Als in een netwerk een record of bestand in gebruik is, krijgt u geen toegang tot dat record of bestand. Met de parameter <Nuitdr> kunt u opgeven hoe vaak moet worden geprobeerd toegang te krijgen. Het bereik dat u kunt opgeven bij de parameter <Nuitdr> loopt van -1 tot en met 32.000. Als u -1 opgeeft, wordt een oneindig aantal keren geprobeerd toegang te krijgen. In de tussentijd kunt u dBASE IV echter niet gebruiken.

Als u 0 opgeeft zonder een ON ERROR-commando, verschijnt de melding **Even wachten a.u.b., iemand anders heeft dit record of bestand vergrendeld**, en wordt een oneindig aantal keren geprobeerd het record of bestand te vergrendelen. Als u wilt, kunt u de pogingen afbreken door op **Esc** te drukken.

Als het opgegeven getal groter is dan nul, wordt er geen foutmelding weergegeven.

Zie ook

FLOCK(), NETWORK(), ON ERROR, RLOCK()/LOCK, SET(), UNLOCK

SET SAFETY

Met SET SAFETY kunt u voorkomen dat u per ongeluk een bestaand bestand overschrijft. In dat geval wordt eerst een melding weergegeven die u vraagt of u werkelijk het bestand wilt overschrijven.

Syntaxis

SET SAFETY ON/off

Standaardinstelling

De standaardinstelling van SET SAFETY is ON.

Gebruik

Als SET SAFETY op ON is ingesteld en u een bestand probeert te maken met dezelfde naam als een bestaand bestand, wordt het volgende foutvenster weergegeven:r

```
Bestand bestaat al
Overschrijven  Annuleren
```

De standaardkeuze is **Overschrijven**.

Druk op ↵ als u het bestaande bestand wilt overschrijven. Het bestaande bestand wordt dan overschreven en alle gegevens in het bestand gaan verloren. Als u het bestand niet wilt overschrijven, plaatst u de keuzebalk op Annuleren en drukt u op ↵. Daarna geeft u een nieuwe naam op voor het te maken bestand.

Als SET SAFETY op OFF is ingesteld, kunt u een bestand overschrijven of gegevens vernietigen zonder dat u wordt gewaarschuwd.

Bij gebruik van de commando's MODIFY COMMAND/FILE en MODIFY STRUCTURE worden automatisch reservekopieën van de originele bestanden gemaakt voordat de commando's daadwerkelijk worden uitgevoerd. Deze commando's negeren de instelling van SET SAFETY bij het creëren van reservekopieën en bestaande reservekopieën worden dus altijd overschreven.

Als u het commando ZAP geeft en SET SAFETY op ON is ingesteld, wordt u gevraagd te bevestigen dat alle records uit het bestand moeten worden gewist.

Zie ook

COPY, COPY FILE, INDEX, JOIN, SAVE, SET(), SORT, TOTAL, UPDATE, ZAP

SET SCOREBOARD

Als SET SCOREBOARD op ON en SET STATUS op OFF is ingesteld, worden de toetsenbordaanduidingen weergegeven op regel 0 van het beeldscherm.

Syntaxis

SET SCOREBOARD ON/off

Standaardinstelling

De standaardinstelling van SET SCOREBOARD is ON.

Gebruik

Als SET SCOREBOARD op ON en SET STATUS op OFF is ingesteld, worden de toetsenbordaanduidingen weergegeven op regel 0 van het scherm. **Del** geeft aan dat een record een wismarkering heeft, **Ins** geeft aan dat de tekst wordt tussengevoegd, **Caps** geeft aan dat Caps Lock is ingedrukt en **Num** geeft aan dat Num Lock is ingedrukt. In een lokaal netwerk wordt een record- of bestandsvergrendeling aangeduid door **RecGren** en **BestGren**.

Als SET STATUS op ON is ingesteld, worden de aanduidingen weergegeven op de statusbalk en heeft SET SCOREBOARD geen effect.

Wanneer zowel SET SCOREBOARD als SET STATUS op OFF zijn ingesteld, worden de aanduidigen niet weergegeven.

Zie ook

SET(), SET MESSAGE, SET STATUS

SET SEPARATOR

Met SET SEPARATOR kunt u het scheidingsteken voor duizendtallen veranderen in een ander teken.

Syntaxis

SET SEPARATOR TO [<Tuitdr>]

Standaardinstelling

Het standaardteken van SET SEPARATOR is de punt.

Gebruik

Het scheidingsteken kan slechts uit één teken bestaan, maar mag geen cijfer of letter zijn. De standaardinstelling kan worden gewijzigd in het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV*.

Voorbeelden

U kunt een getal als volgt weergeven volgens de Amerikaanse standaard, met een komma als scheidingsteken tussen duizendtallen en een punt als decimaalteken:

```
. SET SEPARATOR TO ","
. SET POINT TO "."
. waarde = 100000.55
. ? waarde PICTURE "999,999.99"
100,000.55
```

Zie ook

SET POINT

SET SKIP

Het commando SET SKIP werkt in combinatie met SET RELATION en koppelt database-bestanden als er een enkelvoudige of complexe relatie bestaat tussen records in het ouderbestand en in de kindbestanden. Met dit commando hebt u toegang tot alle records in de kindbestanden die aan de sleutelvoorwaarde voldoen.

Syntaxis

SET SKIP TO [<alias> [, <alias>]...]

Gebruik

Wanneer er in de SKIP-lijst een werkgebied is opgenomen, worden alle kindrecords uit het kindwerkgebied die voldoen aan de sleutel, verwerkt (en niet alleen het eerste). De verwerking wordt niet beïnvloed door de volgorde waarin de werkgebieden in de lijst staan, maar u mag alleen werkgebieden in de lijst opnemen die met SET RELATION zijn gekoppeld. SET SKIP heeft alleen betrekking op kindwerkgebieden. U hoeft het actieve werkgebied dus niet in de lijst op te nemen.

De instelling van SET SKIP heeft invloed op alle commando's waarmee de recordwijzer wordt verplaatst en op commando's met de clausules FOR en WHILE.

Indien er een lijst is opgegeven bij SKIP wordt de recordwijzer voorwaarts verplaatst in het laatste werkgebied uit de kettingrelatie zolang dit werkgebied in de SKIP-lijst voorkomt. Stel dat het actieve werkgebied A is en dat u maar één werkgebied (B) in de SKIP-lijst opneemt. Het commando ziet er als volgt uit:

SET SKIP TO B

Het commando LIST heeft tot gevolg dat de recordwijzer eerst in B wordt verplaatst. Op het moment dat de recordwijzer in B bij een record komt waarvan de relatiesleutel niet meer overeenkomt met het actieve record in A, wordt de recordwijzer in A verplaatst. Op die manier worden eerst alle records in B verwerkt die dezelfde relatiesleutel als het record in A hebben. Pas dan wordt de recordwijzer in A naar het volgende record verplaatst. SET SKIP werkt op dezelfde manier als er drie of meer bestanden zijn gekoppeld. U kunt bij een relatie tussen drie bestanden (A met B, B met C) vanuit werkomgeving A bijvoorbeeld het commando geven:

SET SKIP TO B,C

of

SET SKIP TO C,B

Beide commando's zorgen ervoor dat eerst de records in C die de relatiesleutel bezitten, worden verwerkt. Wanneer er een record wordt aangetroffen dat niet de relatiesleutel bezit, worden de records in werkgebied B verwerkt totdat ook in B geen record meer wordt gevonden dat de relatiesleutel bezit. Pas dan wordt de recordwijzer in A naar een volgend record verplaatst.

Uitgaand van dezelfde relatie tussen A, B en C kunt u vanuit werkgebied A ook het volgende commando geven:

SET SKIP TO C

SET SKIP

Nadat alle records in C met de relatiesleutel zijn verwerkt, wordt de recordwijzer in A verplaatst. In de SET SKIP-lijst hoeft u dus niet alle bestanden uit de kettingrelatie op te nemen.

Voorbeelden

Met SET SKIP kunt u alle overeenkomstige records in gerelateerde database-bestanden weergeven:

```
. SELECT 3
. USE Artikel ORDER Bestel_nr
Hoofdindex: BESTEL_NR
. SELECT 2
. USE Transact ORDER Klant_nr
Hoofdindex: KLANT_NR
. SET RELATION TO Bestel_nr INTO Artikel
. SELECT 1
. USE Afnemers && Hoofd-databasebestand.
. SET RELATION TO Klant_nr INTO Transact
. SET SKIP TO Magazijn, Afnemers, Transact
. LIST Klantnaam, B->Bestel_nr, Magazijn->Artikel
Recordnr.   Klantnaam                 B->Bestel_nr   Magazijn->Artikel
     1      WIJSMAN & ZONEN BV
     2      BALJUW & CO.              87-109         TAFEL,BIJZET
     2      BALJUW & CO.              87-109         LAMP,STAAND
     2      BALJUW & CO.              87-112         STOEL,KANTINE
     3      L. G. BLOEM & PARTNERS    87-106         SOFA,8 POTEN
     3      L. G. BLOEM & PARTNERS    87-108         STOEL,BUREAU
     3      L. G. BLOEM & PARTNERS    87-115         LAMP,STAAND
     4      SAUR & LANGWEER           87-111         STOEL,BUREAU
     5      SMITS & SMITS             87-113         BOEKENKAST
     5      SMITS &SMITS              87-105         SOFA,6 POTEN
     5      SMITS & SMITS             87-105         SOFA,6 POTEN
     6      TIEMESE & KEEFMAN BV      87-107         STOEL,BUREAU
     6      TIEMESE & KEEFMAN BV      87-110         DOSSIERKAST,2 LADEN
     6      TIEMESE & KEEFMAN BV      87-110         DOSSIERKAST,4 LADEN
     7      VOLLEBREGT IMPORT         87-114         LAMP,STAAND
     8      PUNTER PROMOTIONS         87-105         SOFA,6 POTEN
     8      PUNTER PROMOTIONS         87-105         SOFA,6 POTEN
     8      PUNTER PROMOTIONS         87-116         BUREAU,DIRECTIE,5 POTEN
     8      PUNTER PROMOTIONS         87-116         LADENKAST, 2 LADEN
     8      PUNTER PROMOTIONS         87-116         STOEL,BUREAU
```

Bij het eerste record worden de velden Bestel_nr en Artikel niet weergegeven omdat Wijsman & Zonen BV geen bestelling hebben geplaatst. Met het volgende commando worden alleen de klanten weergegeven die een bestelling hebben geplaatst:

```
. LIST Klantnaam, B->Bestel_nr, Magazijn->Artikel FOR "" <> TRIM(B->Bestel_nr
```

Zie ook

SET RELATION

SET SPACE

Met SET SPACE kunt u instellen dat bij de commando's ? en ?? een extra spatie tussen de uitdrukkingen wordt afgedrukt.

Syntaxis

SET SPACE ON/off

Standaardinstelling

De standaardinstelling van SET SPACE is ON omwille van de compatibiliteit met dBASE III PLUS.

Gebruik

Met het commando SET SPACE kunt u een extra spatie plaatsen tussen de velden in de commando's ? en ??. Typ een komma tussen de velden die u wilt scheiden met SET SPACE.

Voorbeeld

```
.naam = "Suzan",
Suzan
. stad = "Boxtel"
Boxtel
. ? naam, stad
Suzan Boxtel
. SET SPACE OFF
. ? naam, stad
SuzanBoxtel
```

Zie ook

SET()

SET SQL

Met SET SQL kunt u de interactieve SQL-modus van dBASE IV activeren. Wanneer SQL eenmaal is geactiveerd, kunnen alleen SQL-commando's en een deel van de commando's van dBASE IV worden gegeven.

Syntaxis

SET SQL on/OFF

Standaardinstelling

De standaardinstelling van SET SQL is OFF.

Gebruik

Het commando SET SQL ON kan worden gegeven vanaf de commandostip. De commandostip wordt dan vervangen door de SQL-aanwijzing en vanaf dat moment kunnen alleen geldige SQL-commando's en een deel van de commando's van dBASE IV worden gegeven.

In de SQL-modus kan het commando SET SQL OFF worden gegeven. De SQL-aanwijzing wordt dan vervangen door de commandostip. Als SQL op OFF is ingesteld, kunnen alleen geldige commando's en functies van dBASE IV worden gegeven.

SQL ON/OFF kan niet worden gebruikt in een programma.

Zie ook

SET()

U kunt dBASE ook starten in SQL-modus. Raadpleeg *Aan de slag met dBASE IV* voor meer informatie over het wijzigen van het bestand Config.db.

SET STATUS

Met SET STATUS kunt u bepalen of de statusbalk wordt weergeven onder aan het scherm wanneer u werkt vanaf de commandostip, vanuit een programma en bij schermgrote bewerkingen zoals EDIT, APPEND, BROWSE en READ.

Syntaxis

SET STATUS on/OFF

Standaardinstelling

De standaardinstelling van SET STATUS is OFF. SET STATUS wordt echter op ON ingesteld in het bestand Config.db dat wordt geladen als dBASE IV individueel wordt gebruikt. Als het bestand Config.db niet wordt geladen, is OFF de standaardinstelling van SET STATUS.

Gebruik

Wanneer SET STATUS op ON is ingesteld, worden de naam van het commando, het actieve bestand en het actieve recordnummer van het totale aantal records op de statusbalk weergegeven. Wanneer SET STATUS op OFF is ingesteld, wordt deze informatie niet weergegeven.

Met SET STATUS wordt ook de weergave van *scoreboard*-informatie bestuurd. Daartoe behoren aanduidingen voor **Del**, **Ins**, **Caps Lock**, **Num Lock**, **BestGren** en **RecGren**. Als SET STATUS op ON is ingesteld, wordt de informatie van SCOREBOARD op de statusbalk weergegeven. Wanneer SET STATUS op OFF en SET SCOREBOARD op ON is ingesteld, wordt de informatie van SCOREBOARD in de rechter bovenhoek van het scherm weergegeven. Als zowel SET STATUS als SET SCOREBOARD op OFF is ingesteld, wordt de informatie van SCOREBOARD niet weergegeven.

Als u een commando geeft waardoor de informatie op de statusbalk niet juist meer is, wordt de statusbalk bijgewerkt.

Tips

Als SET STATUS op ON is ingesteld, moet u geen velden of tekst onder regel 21 (of regel 41 in EGA43-modus) opgeven. Als het deel van het beeldscherm onder regel 21 wordt gebruikt, zullen de statusregel en de informatieregel worden overschreven door uw programma's en gegevens.

Zie ook

SET(), SET MESSAGE, SET SCOREBOARD

SET STEP

Met SET STEP kunt u instellen dat een programma na iedere opdracht wordt onderbroken. Op die manier kunt u een programma stap voor stap uitvoeren. Dit commando wordt hoofdzakelijk gebruikt bij het opsporen van fouten in programma's.

Syntaxis

SET STEP on/OFF

Standaardinstelling

De standaardinstelling van SET STEP is OFF.

Gebruik

Wanneer SET STEP op ON is ingesteld, wordt steeds één opdracht uitgevoerd. Het resultaat van die opdracht wordt weergegeven. Voordat de volgende instructie wordt uitgevoerd, wordt de volgende melding weergegeven:

Spatiebalk voor de volgende stap, S om te onderbreken of Esc om te annuleren...

Het programma wordt onderbroken totdat u een keuze maakt.

Zie ook

COMPILE, DEBUG, DO, SET(), SET DEBUG, SET DEVELOPMENT, SET ECHO, SET TALK

SET TALK

Met SET TALK kunt u bepalen of de antwoorden op bepaalde commando's van dBASE IV worden weergegeven.

Syntaxis

SET TALK ON/off

Standaardinstelling

De standaardinstelling van SET TALK is ON.

Gebruik

Wanneer dBASE IV interactief wordt gebruikt, kunt u SET TALK het beste instellen op ON. Wanneer een record of commando wordt verwerkt als SET TALK op ON is ingesteld, wordt bepaalde informatie weergegeven: recordnummers, geheugenvariabelen en het resultaat van commando's als APPEND FROM, COPY, PACK, STORE en SUM.

In programma's kunt u met SET TALK OFF bepalen wat er op het scherm wordt weergegeven en wat er op de printer wordt afgedrukt.

Wanneer SET TALK op OFF is ingesteld, worden waarschuwingen van de compiler niet weergegeven, terwijl ook de recordteller die bij sommige commando's wordt weergegeven, is uitgeschakeld.

Voorbeeld

Als u SET TALK op OFF instelt, wordt het antwoord van een commando niet langer weergegeven:

```
. Mget=12345
12345
. SET TALK OFF
. Mtek ="Hallo"
.
```

Zie ook

SET(), SET DEBUG, SET ECHO, SET ODOMETER, SET STEP

SET TITLE

Met SET TITLE kunt u de titelaanwijzing van het catalogusbestand in- en uitschakelen.

Syntaxis

SET TITLE ON/off

Standaardinstelling

De standaardinstelling van SET TITLE is ON.

Gebruik

Als SET CATALOG op ON is ingesteld, worden bestanden automatisch toegevoegd aan de catalogus wanneer ze worden gemaakt, gebruikt of opgeslagen. Als SET TITLE op ON is ingesteld en een nieuw bestand in de catalogus wordt opgenomen, wordt u gevraagd een titel te geven voor het bestand. Wanneer SET TITLE op OFF is ingesteld, hoeft u geen titel op te geven.

In **Rapportontwerp**, **Etiketontwerp**, **Database-ontwerp**, **Query-ontwerp**, **Formulierontwerp**, **Programma-ontwerp** en **Toepassingsontwerp** kunt u een titel (ook wel omschrijving genoemd) toevoegen of wijzigen met behulp van de afrolmenu's.

U kunt ook met commando's als EDIT en REPLACE een titel toevoegen aan het catalogusrecord van een bestand. U wijzigt dan het titelveld in het catalogusbestand dat is geopend in werkgebied 10. U kunt in elk werkgebied een catalogusbestand openen en bewerken, maar werkgebied 10 wordt automatisch gekozen als u het commando SET CATALOG geeft.

Zie ook

ASSIST, EDIT, REPLACE, SET(), SET CATALOG

SET TRAP

Met SET TRAP kunt u opgeven dat de debugger moet worden aangeroepen als er op **Esc** wordt gedrukt of als zich een fout in een programma voordoet.

Syntaxis

SET TRAP on/OFF

Standaardinstelling

De standaardinstelling van SET TRAP is OFF.

Gebruik

Wanneer SET TRAP op ON is ingesteld, wordt het foutopsporingsprogramma opgeroepen als zich een fout voordoet of als er op **Esc** wordt gedrukt terwijl SET ESCAPE op ON is ingesteld. Het programma wordt onderbroken op de regel waarop de fout zich voordeed. ON ERROR heeft een hogere prioriteit dan SET TRAP en het programma wordt bestuurd door ON ERROR.

Wanneer SET TRAP op OFF is ingesteld en er geen voorwaarde is opgegeven met ON ERROR, heeft u drie keuzemogelijkheden als zich een fout voordoet:

- Annuleer het programma
- Negeer de fout
- Onderbreek het programma

Als SET TRAP op ON is ingesteld, wordt de debugger aangeroepen.

Zie ook

DEBUG, ON ERROR, SET(), SET DEVELOPMENT, SET ESCAPE

SET TYPEAHEAD

Met SET TYPEAHEAD kunt u de grootte van de typebuffer instellen.

Syntaxis

SET TYPEAHEAD TO <Nuitdr>

Standaardinstelling

Standaard worden 20 tekens opgeslagen in de typebuffer. Het bereik dat kan worden opgegeven, loopt van 0 tot en met 32.000.

Gebruik

Als u snel typt, kunt u met dit commando de capaciteit van de typebuffer vergroten zodat er geen toetsaanslagen verloren gaan als u een voorsprong neemt op het programma. SET TYPEAHEAD werkt alleen als SET ESCAPE op ON is ingesteld. Met SET TYPEAHEAD bepaalt u ook hoeveel tekens er met het commando KEYBOARD in de typebuffer kunnen worden opgeslagen.

Bij de schermgrote bewerkingen EDIT en APPEND is de typebuffer altijd 20 tekens groot, ongeacht de instelling van SET TYPEHEAD.

Tips

Als u een foutverwerkingprogramma uitvoert met ON ERROR DO <programma>, kunt u het beste de typebuffer op 0 instellen. Neem het commando SET TYPEAHEAD TO 0 op als eerste regel van het foutverwerkingprogramma. Hierdoor worden het commando ON KEY en de functie INKEY() buiten werking gesteld. Omdat een fout een onverwachte situatie is, wordt de typebuffer als voorzorgsmaatregel buiten werking gesteld.

Speciale gevallen

Als u meer tekens typt dan er in de typebuffer passen, gaan alle extra tekens verloren. Als SET BELL op ON is ingesteld, hoort u in dat geval een pieptoon. Als dit vaak gebeurt, is het verstandig de typebuffer groter te maken.

Zie ook

INKEY(), KEYBOARD, ON, SET(), SET BELL, SET ESCAPE

SET UNIQUE

Met SET UNIQUE kunt u bepalen of alle records die dezelfde sleutelwaarde hebben, moeten worden opgenomen in het indexbestand (.ndx) of het samengestelde indexbestand (.mdx).

Syntaxis

SET UNIQUE on/OFF

Standaardinstelling

De standaardinstelling van SET UNIQUE is OFF.

Gebruik

Als SET UNIQUE op ON is ingesteld en u een nieuwe index maakt voor een database-bestand waarin meerdere records dezelfde sleutelwaarde hebben, wordt alleen het eerste record met die sleutelwaarde opgenomen in het nieuwe indexbestand. Het indexbestand is dan *uniek*.

Als u een uniek indexbestand opnieuw indexeert met REINDEX, blijft het bestand uniek, ongeacht de instelling van SET UNIQUE.

Unieke indexen worden in dBASE IV maar eenmaal verwerkt. Daarom wordt een sleutelwaarde die eerder verborgen is, niet automatisch bijgewerkt wanneer de sleutelwaarden door een bepaalde bewerking veranderen. Met REINDEX worden nadrukkelijk alle waarden in een unieke index bijgewerkt.

Als er een unieke index is geopend voor een database-bestand en u nieuwe records toevoegt of bestaande records bewerkt, verandert dat niets aan de unieke status van het indexbestand. Als u dus met APPEND een record toevoegt waarvan de indexsleutel al in de index voorkomt, wordt dat record niet opgenomen in de index, hoewel het wel wordt toegevoegd aan het bestand.

Als u de indexsleutel van een record wijzigt en de nieuwe sleutel al in de index voorkomt, wordt dat record uit de index verwijderd.

Richtlijnen voor het programmeren

U kunt een unieke index niet-uniek maken door het indexbestand opnieuw te maken met het commando INDEX ON terwijl SET UNIQUE op OFF is ingesteld.

Wanneer in een unieke index een record dezelfde sleutelwaarde heeft als een ander record, blijft dat andere record verborgen. Wanneer nu de indexsleutel van het record in de unieke index wordt gewijzigd, wordt het verborgen record pas opgenomen in de index als de index opnieuw wordt geïndexeerd met REINDEX.

SET UNIQUE

Voorbeeld

Op de volgende manier kunt u een lijst maken waarin de inhoud van een veld wordt weergegeven, zonder dat de inhoud van een veld twee keer voorkomt:

```
. USE Magazijn
. SET UNIQUE ON
. INDEX ON Artikel TO Artnaam
100% geïndexeerd  10 records geïndexeerd
. LIST Artikel

Recordnr.      Artikel
       12      BOEKENKAST
       15      BUREAU,DIRECTIE,5 POTEN
        8      DOSSIERKAST,2 LADEN
        9      DOSSIERKAST,4 LADEN
        7      LAMP,STAAND
        1      SOFA,6 POTEN
        3      SOFA,8 POTEN
        4      STOEL,BUREAU
       11      STOEL,KANTINE
        6      TAFEL,BIJZET
```

Zie ook

FIND, INDEX, REINDEX, SEEK, SEEK(), SET(), SET INDEX, SET ORDER, USE

SET VIEW

Met SET VIEW kunt u een query (.qbo of .qbe) uitvoeren die is gemaakt in het Control Center of vanaf de commandostip. Bovendien kunt u met SET VIEW de omgeving in een visiebestand van dBASE III PLUS bijwerken.

Syntaxis

SET VIEW TO <naam query-bestand>/?

Gebruik

Met het commando SET VIEW kunt u een query van dBASE IV uitvoeren die is gedefinieerd in een .qbo- of .qbe-bestand. U kunt een query maken door het query-scherm op te roepen vanuit het Control Center of het commando CREATE/MODIFY QUERY of CREATE/MODIFY VIEW te geven.

U kunt de naam van een visiebestand opgeven bij het commando of met de vraag-clausule, (?), een menu oproepen met alle .qbo-, .qbe- en .vue-bestanden in de geopende catalogus. Als er geen catalogus is geopend, wordt een lijst weergegeven met alle bestanden in de actieve bestandsindex.

Wanneer er een catalogus is geopend als u het commando SET VIEW TO geeft, wordt de catalogus bijgewerkt. Als een catalogus is geopend en SET CATALOG op ON is ingesteld, wordt ieder bestand dat wordt geopend, toegevoegd aan de catalogus als het er nog niet instaat. Als SET TITLE op ON is ingesteld, wordt u gevraagd een titel op te geven voor het bestand dat aan de catalogus wordt toegevoegd.

Zoekvolgorde

Wanneer u het commando SET VIEW TO geeft, zoekt dBASE IV eerst naar een .qbo-bestand. Als dBASE IV geen .qbo-bestand kan vinden in de op dat moment ingestelde paden, wordt er gezocht naar een .qbe-bestand. Een .qbe-bestand is een query-bestand dat nog niet is gecompileerd. Als dBASE IV het .qbe-bestand vindt, wordt dat bestand gecompileerd en uitgevoerd.

Als u een .vue-bestand van dBASE III PLUS gebruikt, maakt dBASE IV een .qbe-bestand van het .vue-bestand. Als u daarna dit .qbe-bestand gebruikt, wordt het .qbe-bestand gecompileerd naar een .qbo-bestand en wordt dit bijgewerkte visiebestand gebruikt.

Een visiebestand van dBASE IV (.qbo of .qbe) kan niet worden gebruikt in dBASE III PLUS.

Visiebestanden van dBASE IV

In een visiebestand van dBASE IV zijn de volgende elementen opgenomen:

- alle geopende database-bestanden, indexbestanden en het nummer van het werkgebied van elk bestand.
- alle relaties tussen de database-bestanden

SET VIEW

- het nummer van het actieve werkgebied
- de actieve veldenlijst
- het actieve filter
- het geopende indelingsbestand (.fmt), indien aanwezig

Zie ook

CREATE/MODIFY VIEW/QUERY, SET(), SET CATALOG, SET FIELDS, SET FILTER, SET FORMAT, SET INDEX, SET ORDER, SET RELATION, SET TITLE

SET WINDOW

Met SET WINDOW kunt u een standaardvenster instellen waarin u een memoveld kunt bewerken terwijl u bezig bent met APPEND, BROWSE, CHANGE, EDIT of READ.

Syntaxis

SET WINDOW OF MEMO TO <naam venster>

<naam venster> is de naam van een eerder gedefinieerd venster.

Gebruik

Met het commando SET WINDOW kunt u in een eerder gedefinieerd venster memovelden bewerken terwijl u met een schermgroot commando bezig bent, bijvoorbeeld APPEND, BROWSE, CHANGE, EDIT of READ.

De clausule WINDOW van het commando @...GET heeft een hogere prioriteit dan het commando SET WINDOW bij het bewerken van memovelden.

Voorbeeld

Bij het eerste en derde @...GET-commando worden de memo's bewerkt in het venster dat als Johan is gedefinieerd. Bij het tweede @...GET-commando worden de memo's bewerkt in het venster dat is gedefinieerd als Marjan

```
. DEFINE WINDOW JOHAN FROM 5,5 TO 15,60
. DEFINE WINDOW MARJAN FROM 1,30 TO 11,70
. SET WINDOW OF MEMO TO JOHAN
. @ 1,1 GET MEMO1
. @ 2,1 GET MEMO2 WINDOW MARJAN
. @ 3,1 GET MEMO3
. READ
```

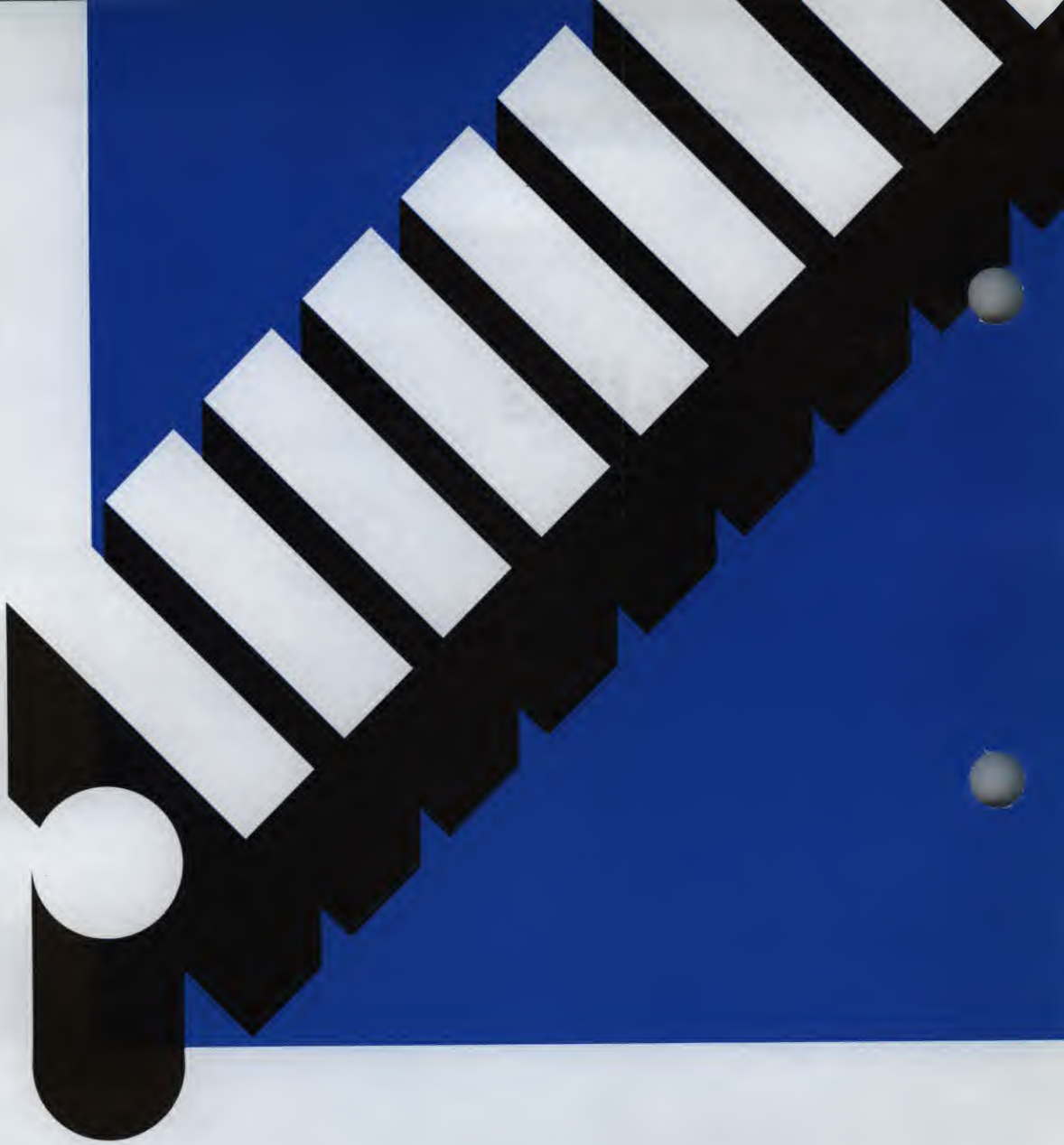
Als u een ongedefinieerd venster opgeeft bij SET WINDOW OF MEMO TO, verschijnt de foutmelding *Venster is niet gedefinieerd* op het scherm.

Zie ook

ACTIVATE SCREEN, CLEAR WINDOW, DEFINE WINDOW, MOVE WINDOW

Naslagshandboek van dBASE IV

Functies



Functies

4

&

& is de macrosubstitutiefunctie. Op sommige plaatsen gebruikt dBASE IV de naam van een geheugenvariabele letterlijk. & vervangt in die gevallen de naam van de variabele door de inhoud. Deze functie kan alleen worden gebruikt voor tekenvariabelen.

Syntaxis

& <tekenvariabele> [.]

Gebruik

Met de functie & kunt u de inhoud van een geheugenvariabele van het type teken opvragen wanneer dBASE IV een constante waarde verwacht in plaats van een tekenuitdrukking. FIND en USE zijn voorbeelden van commando's waarbij dBASE IV een constante waarde verwacht, net als andere commando's waarbij een bestandsnaam wordt opgegeven.

De punt (.) is een optionele macrostop. Wanneer een macro wordt gebruikt als prefix van een constante tekenreeks, geeft een macrostop het einde van de macro aan.

U kunt in macro's geen gestructureerde programmeercommando's gebruiken omdat in dBASE IV gedurende de uitvoering van het programma geen macro's kunnen worden gecompileerd. Het betreft hier DO WHILE-lussen, DO CASE-constructies, IF-constructies, SCAN...ENDSCAN, TEXT...ENDTEXT, PRINTJOB...ENDPRINTJOB en BEGIN/END TRANSACTION. U kunt daarentegen wel macro's gebruiken in het voorwaarde-element van de commando's IF, CASE, DO WHILE en SCAN.

U kunt de macrosubstitutiefunctie niet gebruiken voor een database-tekenveld.

Aangezien de macrosubstitutiefunctie in een DO WHILE-voorwaarde maar eenmaal wordt geëvalueerd, moet u de &-macro alleen gebruiken als de waarde ervan binnen de DO WHILE-lus niet wijzigt.

Voorbeelden

U kunt geheugenvariabelen opgeven bij het commando FIND:

```
. Mnaam = "Peters"
Peters
. USE Afnemers INDEX KIntnaam
. FIND &Mnaam.
. ? RECNO(), Achternaam
                    4 Peters
```

In het volgende voorbeeld wordt de bestandsnaam van een database toegewezen aan een geheugenvariabele. Vervolgens wordt bij het commando USE de variabele opgegeven in plaats van de bestandsnaam.

```
. ACCEPT "Geef de bestandsindex van het databasebestand op: " TO Mbestindex
Geef de bestandsindex van het databasebestand op: \dBASE
. ACCEPT "Geef een bestandsnaam op: " TO bestnaam
Geef een bestandsnaam op: Factuur
. USE C:&Mbestindex.\&bestnaam.
. ? DBF()
C:\DBASE\FACTUUR.DBF
```

Door een nieuwe aanpak worden in dBASE IV kunt u ook indirecte bestandsverijzingen gebruiken in plaats van macrosubstitutie. Bij deze methode wordt de compiler niet aangeroepen en wordt het commando dus sneller uitgevoerd:

```
. USE "C:"+Mbestindex+:"\"+bestnaam
```

Raadpleeg het gedeelte over bestandsnamen in hoofdstuk 1 voor meer informatie over indirecte bestandsnamen en verwijzingen.

ABS()

Met de functie ABS() kunt u de absolute waarde van een numerieke uitdrukking opvragen ongeacht het gebruikte teken.

Syntaxis

ABS(<Nuitdr>)

Gebruik

Met de functie ABS() kunt u de absolute waarde van een numerieke uitdrukking opvragen. De absolute waarde van zowel +3 als -3 is 3. Met deze functie kunt u het verschil tussen twee getallen berekenen zonder dat rekening wordt gehouden met het teken.

De waarde die door de functie wordt gegeven, is altijd positief.

Voorbeeld

Op de volgende manier kunt u het aantal dagen tussen twee datums bereken:

```
. datum1 = {25-12-88}
25-12-88
. datum2 = {01-04-88}
01-04-88
. ? ABS(datum1 - datum2)
                    268.00
. ? ABS(datum2 datum1)
                    268.00
```

ACCESS()

Met de functie ACCESS() kunt u het toegangsniveau van de actieve gebruiker opvragen.

Syntaxis

ACCESS()

Gebruik

Met de functie ACCESS() kunt u een toepassingsprogramma beveiligen in een netwerk. Aan de hand van het toegangsniveau dat deze functie oplevert, kunt u nagaan welke rechten zijn toegekend met PROTECT.

Richtlijnen voor het programmeren

Met de functie ACCESS() kunt u in een programma toegangsrechten wijzigen.

```
IF ACCESS() < 3
```

kan bijvoorbeeld eenvoudig worden gewijzigd in:

```
IF ACCESS() < 8
```

Controleer het toegangsniveau altijd met een gecompileerd programma, om te voorkomen dat een gebruiker zijn of haar toegangsniveau aanpast. Verwijder het bronbestand (.prg) uit het toegangsniveau van de gebruiker. De gecompileerde programmabestanden kunnen niet door de gebruiker worden gewijzigd.

In het volgende voorbeeldprogramma wordt het toegangsniveau van de gebruiker gecontroleerd. Als het toegangsniveau lager is dan 3, wordt een subprogramma uitgevoerd. Als het toegangsniveau 3 of hoger is, hoort u een pieptoon en wordt de melding **Geen toegangsrecht op dit niveau** weergegeven.

```
ACCEPT "Maak uw keuze " TO Keuze
   DO CASE
      CASE Keuze="1"
         IF ACCESS() < 3
            DO Program
         ELSE
            ? CHR(7)
            ? "Geen toegangsrecht op dit niveau"
            WAIT
            RETURN TO MASTER
         ENDIF
      CASE Keuze="2"
         *
         *
         *
   ENDCASE
```

Speciale gevallen

Als het bestand Dbsystem.db niet wordt gevonden wanneer dBASE IV wordt gestart, wordt het aanmeldscherm niet weergegeven. Als een gebruiker dBASE IV niet via het aanmeldscherm is binnengekomen, levert de functie ACCESS() nul op. Een gebruiker met toegangsniveau nul heeft geen toegang tot gecodeerde bestanden. Houd het bestand Dbsystem.db in dezelfde bestandsindex als dBASE IV en start dBASE IV altijd via het aanmeldscherm om te voorkomen dat de functie ACCESS() nul oplevert.

Als u programma's schrijft waarin gecodeerde bestanden worden gebruikt, is het verstandig het toegangsniveau van de gebruiker in een vroeg stadium te controleren. Als de functie ACCESS() nul oplevert, kunt u de gebruiker via het programma vragen zich opnieuw aan te melden of contact op te nemen met de netwerkbeheerder.

Wanneer dBASE IV niet in een netwerk wordt gebruikt, levert ACCESS() altijd een nul op.

Zie ook

FLOCK(), NETWORK(), PROTECT, RLOCK(), UNLOCK(), USER()

ACOS()

Met de arccosinusfunctie ACOS() kunt u de hoek in radialen van iedere gegeven cosinus berekenen.

Syntaxis

ACOS(<Nuitdr>)

Gebruik

De variabele (<Nuitdr>) is een numerieke uitdrukking die de cosinus is van een bepaalde hoek. De waarde van de numerieke uitdrukking mag niet kleiner zijn dan -1 en niet groter dan +1.

De uitkomst is altijd een hoek in radialen tussen nul en Pi. Met de commando's SET DECIMALS en SET PRECISION kunt u het aantal decimalen en de nauwkeurigheid bepalen.

Voorbeeld

In het volgende voorbeeld wordt de hoek in radialen, die overeenkomt met een in vier stappen berekende cosinuswaarde, toegewezen aan een geheugenvariabele:

```
. SET DECIMALS TO 4
. x = 8.4852
                    8,4852
. y = 12
                    12
. Mcos = x / y
                    0,7071
. hoek = ACOS(Mcos)
                    0,7854
```

Zie ook

ASIN(), ATAN(), ATN2(), COS(), DTOR(), FIXED(), FLOAT(), PI(), RTOD(), SET DECIMALS, SET PRECISION, SIN(), TAN()

ALIAS()

Met de functie ALIAS() kunt u de aliasnaam van een bepaald werkgebied opvragen. Als u geen werkgebied opgeeft, wordt de naam van het actieve werkgebied weergegeven.

Syntaxis

ALIAS([<Nuitdr>])

Gebruik

Met de functie ALIAS() kunt u de bestandsnaam of unieke alias van elk werkgebied opvragen. Met de optionele variabele <Nuitdr> kunt u een bepaald werkgebied opgeven. Alleen de getallen 1 tot en met 10 zijn geldige numerieke uitdrukkingen. Als u geen werkgebied opgeeft, wordt uitgegaan van het actieve werkgebied.

Als u bij <Nuitdr> een decimaal getal opgeeft, bijvoorbeeld 7,5, wordt het getal afgerond op 7.

ASC()

Met de functie ASC() kunt u de decimale ASCII-code van het eerste teken in een tekenuitdrukking opvragen.

Syntaxis

ASC(<Tuitdr>)

Gebruik

Bijlage G bevat een lijst met de decimale ASCII-waarden van alle tekens in de IBM-tekenset. Gebruik deze lijst om de decimale waarden die deze functie oplevert, te identificeren.

Voorbeelden

```
. ? ASC("Nesten")
                    78.

. Getal = "123"
                    123

. ? ASC(getal)
                    49
```

ASIN()

Met de arcsinusfunctie ASIN() kunt u de hoek in radialen van elke gegeven sinus berekenen.

Syntaxis

ASIN(<Nuitdr>)

Gebruik

De variabele (<Nuitdr>) is een numerieke uitdrukking die de sinus is van een bepaalde hoek. De waarde van de numerieke uitdrukking mag niet kleiner zijn dan -1 en niet groter dan +1.

De uitkomst is altijd een getal met een drijvend decimaalteken dat de hoek in radialen weergeeft tussen -Pi/2 en +Pi/2. Met de commando's SET DECIMALS en SET PRECISION kunt u het aantal decimalen en de nauwkeurigheid bepalen.

Voorbeelden

In het volgende voorbeeld wordt de hoek in radialen berekend van een sinus:

```
. ? ASIN(.5000)
                    ,5236
```

U kunt ook de arcsinus van een uitdrukking berekenen:

```
. ster = .5000
                    0,5000
. X = ASIN(ster)
                    0,5236
```

Zie ook

ACOS(), ATAN(), ATN2(), COS(), DTOR(), RTOD(), SET DECIMALS, SET PRECISION, SIN(), TAN()

AT()

Met de functie AT() kunt u een getal opvragen dat aangeeft waar een tekenreeks begint in een tweede tekenreeks.

Syntaxis

AT(<Tuitdr>,<Tuitdr>/<memoveld>)

Gebruik

De binnenste tekenreeks wordt een subtekenreeks genoemd. Wanneer de subtekenreeks niet wordt gevonden in de tweede uitdrukking, levert de functie een nul op.

Voorbeeld

Zoek in het veld Achternaam van de database Afnemers naar de naam Wijsman:

```
. USE Afnemers
. ? AT("Wijsman", Achternaam) <> 0
.T.
. SKIP
Afnemers: Recordnr. 2
. ? AT("Wijsman", Achternaam) <> 0
.F.
```

Zie ook

LEFT(), RIGHT(), SUBSTR()

ATAN()

Met de arctangensfunctie ATAN() kunt u de hoek in radialen van elke gegeven tangens berekenen.

Syntaxis

ATAN(<Nuitdr>)

Gebruik

De variabele (<Nuitdr>) is een numerieke uitdrukking die de tangens van een bepaalde hoek weergeeft. De waarden die de functie oplevert liggen tussen +Pi/2 tot en met -Pi/2. De uitkomst is altijd een getal dat de hoek in radialen weergeeft. Met de commando's SET DECIMALS en SET PRECISION kunt u het aantal decimalen en de nauwkeurigheid bepalen.

Voorbeelden

In het volgende voorbeeld wordt de hoek in radialen van een tangens berekend:

```
. ? ATAN(1.000)
          0,7854
```

U kunt ook de arctangens van een uitdrukking berekenen:

```
. ster = 1.000
. mvar = ATAN(ster)
          0,7854
```

Zie ook

ACOS(), ASIN(), ATN2(), COS(), DTOR(), RTOD(), SET DECIMALS, SET PRECISION, SIN(), TAN()

ATN2()

Met de arctangensfunctie ATN2() kunt u de hoek in radialen berekenen wanneer zowel de cosinus als de sinus van een bepaald punt worden opgegeven.

Syntaxis

ATN2(<Nuitdr1>,<Nuitdr2>)

<Nuitdr1> is de sinus van een bepaalde hoek en <Nuitdr2> is de cosinus van dezelfde hoek. De waarde van de uitdrukking <Nuitdr1>/<Nuitdr2> moet liggen tussen +Pi en -Pi.

Gebruik

Met de functie ATN2() kunt u de waarde in alle vier kwadranten berekenen. De functie is equivalent aan ATAN(x/y). U kunt beter deze functie gebruiken dan ATAN(x/y), omdat een deling door nul dan wordt voorkomen.

De uitkomst is altijd een getal dat de hoek weergeeft in radialen tussen +Pi en -Pi. Met het commando SET DECIMALS kunt u de nauwkeurigheid bepalen.

Voorbeeld

In dit voorbeeld worden verscheidene meetkundige functies toegepast. De sinus en cosinus van een hoek van 30 graden worden berekend, waarbij de graden worden omgezet in radialen met de functie DTOR(). De uitkomst wordt tijdelijk opgeslagen in de geheugenvariabelen x en y. Daarna wordt binnen de functie RTOD() met de functie ATN2() de hoek berekend die hoort bij de sinus en cosinus. De berekeningen worden uitgevoerd met geheugenvariabelen omdat dBASE IV tot op 20 cijfers nauwkeurig is. U hoeft op deze manier geen lange getallen in te typen en de instelling van SET DECIMALS niet te wijzigen.

```
. x=SIN(DTOR(30))
        0,50
. y=COS(DTOR(30))
        0,87
. ? RTOD(ATN2(x,y))
        30
```

Zie ook

ATAN(), COS(), DTOR(), RTOD(), SET DECIMALS, SIN(), TAN()

BAR()

Met de functie BAR() kunt u het balknummer opvragen van de laatst gekozen balk in een pop-up menu.

Syntaxis

BAR()

Gebruik

Met de functie BAR() kunt u het balknummer opvragen van de laatst gekozen balk in het actieve pop-up menu.

In de volgende situaties levert de functie BAR() nul op:

- Er is geen pop-up menu actief
- Er is geen pop-up menu gedefinieerd
- Er is op **Esc** gedrukt om het pop-up menu inactief te maken

Wanneer het pop-up menu is gedefinieerd met het commando DEFINE BAR, levert de functie BAR() het balknummer (bijvoorbeeld 1 of 2) op.

Als de aanwijzingen van een pop-up zijn gedefinieerd met het commando DEFINE POPUP, is de uitkomst van de functie BAR() het regelnummer van deze aanwijzing. De eerste regel in het pop-up menu is regel 1.

Voorbeeld

In een procedure van een programmabestand kunt u met BAR() controleren welke keuze is gemaakt in het pop-up menu.

```
PROCEDURE Visiproc
   DO CASE
      CASE BAR() = 1
            DO Toev_rec
      CASE BAR() = 2
            DO Bewk_rec
      CASE BAR() = 3
            DO Wis_rec
      CASE BAR() = 4
            DEACTIVATE POPUP
   ENDCASE
RETURN
```

Zie ook

ACTIVATE POPUP, DEFINE POPUP, POPUP(), PROMPT()

BOF()

De functie BOF() geeft het begin van een bestand aan.

Syntaxis

BOF([<alias>])

De optionele alias kan een numerieke uitdrukking zijn tussen de 1 en de 10 of een tekenuitdrukking tussen de letters A en J.

Gebruik

Deze functie is bedoeld voor programma's waarin een database in omgekeerde volgorde wordt gelezen. De logische waarde waar (.T.) is het resultaat wanneer wordt geprobeerd de recordwijzer vóór het eerste logische record van het opgegeven bestand te plaatsen.

Wanneer u SKIP -1 opgeeft terwijl de recordwijzer op het eerste bestand staat, levert BOF() waar (.T.) op, waarmee wordt aangegeven dat de recordwijzer aan het begin van het bestand staat. RECNO() blijft echter op 1 staan.

Speciaal geval

Als er geen database actief is, levert BOF() de logische waarde onwaar (.F.) op.

Voorbeeld

Het volgende voorbeeld is een procedure waarmee records kunnen worden bewerkt. De recordwijzer wordt verplaatst aan de hand van de ingedrukte toets. Met de functie BOF() kunt u voorkomen dat de recordwijzer naar een niet bestaand record wordt verplaatst.

```
DO WHILE .T.
   EDIT NEXT 1
   DO CASE
      CASE STR(READKEY(),3) $ 6, 262
         * ——PgUp-toets is ingedrukt.
         SKIP 1
         * ——EDIT niet mogelijk vóór BOF().
         IF BOF()
                                GO TOP
         ENDIF
      *
      *
   ENDCASE
ENDDO
```

Zie ook

EOF(), RECNO(), SKIP

CALL()

Met de functie CALL() kunt u tijdens het programmeren binaire programmamodules uitvoeren.

Syntaxis

CALL (<Tuitdr>,<uitdr>,[<lijst uitdrukkingen>])

<Tuitdr> is een binaire bestandsnaam zonder extensie.

Gebruik

Met de functie CALL() kunt u vanuit programma's van dBASE IV binaire programma's uitvoeren die in een andere taal zijn geschreven (bijvoorbeeld assembleertaal of C). De subroutine wordt eerst geladen met het commando LOAD en daarna uitgevoerd met de functie CALL(). De waarde die de functie oplevert, kan worden gebruikt in een uitdrukking van dBASE IV.

Raadpleeg in hoofdstuk 2 het gedeelte over het commando LOAD voor bijzonderheden over het schrijven van een binair programma.

Er kunnen geheugenvariabelen en reekselementen worden doorgegeven aan de binaire programmamodule. U moet geheugenvariabelen eerst definiëren voordat deze kunnen worden doorgegeven met de functie CALL(). Reeksen moeten eerst worden gedefinieerd, waarna de coördinaten van de elementen kunnen worden opgegeven bij CALL(). U kunt maximaal zeven parameters doorgeven aan de programmamodule.

CALL() accepteert acht totale argumenten. De eerste waarde is altijd de naam van de module, de zeven resterende zijn parameters voor het programma. Aangezien een functie maar één waarde kan opleveren, levert alleen de tweede waarde die is doorgegeven aan CALL(), een uitkomst op. De overige variabelen worden wel bijgewerkt in het geheugen, maar leveren in de functie CALL() geen waarde op.

Voorbeeld

In het volgende voorbeeldprogramma wordt het binaire bestand in het geheugen geladen met het commando LOAD. Het bestand Getdrive.asm staat op een van de voorbeelddiskettes. Daarna wordt de reeks Mstation gedefinieerd. Het eerste element van Mstation krijgt een spatie toegewezen. De resterende negen elementen blijven nul. Met de eerste ?-opdracht wordt de gebruiker verteld wat het standaard station is. Met de tweede ?-opdracht wordt gecontroleerd of het eerste element van Mstation ook werkelijk C is.

```
. LOAD Getdrive
. DECLARE Mstation[10]
. Mstation[1] = " "
. ? "Het actieve station is", CALL(Getdrive,Mstation[1])
Het actieve station is C
. ? Mdrives[1]
C
```

Zie ook

CALL, DECLARE, LOAD, RUN, RUN(), STORE

CDOW()

Met de functie CDOW() kunt u opvragen op welke dag van de week een bepaalde datumuitdrukking valt.

Syntaxis

CDOW(<Duitdr>)

Gebruik

De datumuitdrukking kan zijn: een geheugenvariabele of een veld, of een functie die een datum oplevert.

Voorbeelden

```
. ? CDOW({29-02-88})
maandag
```

In het volgende voorbeeld wordt bepaald of het vandaag een doordeweekse dag is of weekend:

```
. ? "Vandaag is het" + IIF(CDOW(DATE()) = "S", "weekend.", "een doordeweekse dag.")
Vandaag is het een doordeweekse dag.
```

Zie ook

CTOD(), DATE(), DAY(), DOW(), DTOC()

CEILING()

Met de functie CEILING() kunt u het kleinste gehele getal berekenen dat groter dan of gelijk is aan de waarde van de numerieke uitdrukking.

Syntaxis

CEILING(<Nuitdr>)

Gebruik

Met de functie CEILING() kunt u het kleinste gehele getal berekenen dat groter dan of gelijk is aan een gegeven waarde. De resulterende waarde is van hetzelfde type als de opgegeven numerieke uitdrukking.

Voorbeelden

```
. eerste = 123
                    123
. tweede = 10
                    10
. ? CEILING(eerste / tweede)
                    13
```

Met CEILING() berekent u voor negatieve getallen altijd het gehele getal dat het dichtst bij nul ligt. Dat is niet het geval met ROUND().

```
. ? CEILING(5.556)
                    -5,00
. ? ROUND(-5.556,0)
                    -6,00
```

Zie ook

FLOOR(), INT(), ROUND()

CERROR()

De functie CERROR() levert het nummer op van de laatst gemelde compileerfout.

Syntaxis

CERROR()

Standaardinstelling

CERROR() levert een nul op als er geen compileerfouten zijn gevonden.

Gebruik

De functie CERROR() wordt bijgewerkt zodra de dBASE IV-compiler een bewerking uitvoert. De compiler kan worden gestart met het commando COMPILE. De compiler is ook werkzaam na de commando's DO, DEBUG en SET PROCEDURE, door het maken van rapporten, formulieren, etiketten, queries en toepassingen, en door het geven van commando's vanaf de commandostip.

Treedt er een compileerfout op, dan levert ERROR() de waarde **360** op, en MESSAGE() de melding **Compileerfout**. Met CERROR() kunt u het nummer van de betreffende compileerfout opvragen.

Een overzicht van de foutmeldingen is opgenomen in bijlage A van dit boek. Meldingen die CERROR() kan opleveren, zijn in de bijlage gemarkeerd met een asterisk. CERROR() staat los van de waarschuwingen die tijdens het compileren kunnen worden weergegeven zoals:

Commando niet bruikbaar in dBASE IV	**[93]**
Extra tekens aan het einde van commando genegeerd	**[151]**
Commando niet geldig in RunTime-omgeving	**[348]**
Commando is onbereikbaar	**[369]**

Voordat u een nieuw programma uitvoert kunt u de functie CERROR() gebruiken om vast te stellen of de broncode correct is gecompileerd.

Voorbeeld

In het volgende voorbeeld is een stuk uit een programma te zien waarin de functie CERROR() wordt gebruikt in een DO WHILE-lus. Daardoor is de gebruiker gedwongen het programma te bewerken totdat het correct gecompileerd is.

```
DO WHILE .T.
   CLEAR
   MODIFY COMMAND User.prg
   ON ERROR ? ERROR(), MESSAGE()
   COMPILE User.prg
   ON ERROR
   IF CERROR() > 0
      ?
      WAIT "Uw programma is niet gecompileerd. Druk op een toets om defout te corrigeren"
      LOOP
   ENDIF
   EXIT
ENDDO
DO User
```

Zie ook

COMPILE, DEBUG, DO, ERROR(), MESSAGE(), ON ERROR, SET PROCEDURE

CHANGE()

Met de functie CHANGE() kunt u controleren of een record door een andere gebruiker is gewijzigd nadat het is geopend.

Syntaxis

CHANGE([<alias>])

Gebruik

Deze functie werkt alleen bij database-bestanden die zijn bijgewerkt met het commando CONVERT. Een dergelijk database-bestand bevat namelijk een veld genaamd _dbaselock waarin een teller is opgenomen die door dBASE IV wordt bijgewerkt zodra een record wordt gewijzigd.

De functie CHANGE() vergelijkt de waarde van de teller die op het werkstation onder _dbaselock is opgeslagen met de waarde van de teller op de centrale server. Bij verschil tussen beide tellers levert de functie waar (.T.) op. Als beide tellers gelijk zijn levert CHANGE onwaar (.F.) op.

U kunt de waarde van CHANGE() op onwaar instellen door de recordwijzer te verplaatsen. Door het commando GOTO RECNO() wordt het veld _dbaselock van het actieve record opnieuw gelezen. Een daaropvolgende uitvoering van CHANGE() levert onwaar (.F.) op, tenzij het record in de tussentijd door een andere gebruiker is gewijzigd.

Het is mogelijk door toevoeging van een alias records in een ander werkgebied te testen.

OPMERKING
Wanneer een record tijdens een transactie wordt gewijzigd, wordt de teller meteen bijgewerkt. De functie CHANGE levert dan waar (.T.) op. Wanneer de transactie echter niet wordt afgemaakt en de wijzigingen met ROLLBACK ongedaan worden gemaakt, is de uitkomst van de functie niet correct meer.

Zie ook

BEGIN/END TRANSACTION, COMPLETED(), CONVERT, FLOCK(), LKSYS(), RLOCK(), ROLLBACK, ROLLBACK(), SET REFRESH

CHR()

Met de functie CHR() kunt u een numerieke uitdrukking omzetten in een teken.

Syntaxis

CHR(<Nuitdr>)

Gebruik

Met de functie CHR() kunt u elke waarde uit de IBM-tekenset naar het scherm of een printer zenden. <Nuitdr> is een geheel getal tussen 1 en 255.

De IBM-tekenset wordt niet door alle printers ondersteund. Controleer de tabel met ASCII-codes van uw printer als u een teken wel kunt weergeven op het scherm, maar niet kunt afdrukken.

Met de functie CHR() kunt u ook stuurcodes naar de printer zenden. Wanneer u de functie gebruikt in combinatie met de geheugenvariabelen _pscode en _pecode, moet u CHR() niet tussen aanhalingstekens of andere scheidingstekens plaatsen. Zo is bijvoorbeeld

_pscode = CHR(27)+"E"

correct geformuleerd, maar

_pscode = "CHR(27)+E"

op foutieve wijze.

U kunt de functie CHR() niet in afdrukmenu's gebruiken om start- en stopcodes naar de printer te sturen.

Voorbeelden

U kunt op de volgende manier een hoofdletter A (ASCII-code 65) weergeven:

```
. ? CHR(65)
A
```

Met ASCII-code 7 kunt u een pieptoon laten horen:

```
. ? CHR(7)+"Pas op"
Pas op
```

CHR()

Wanneer CHR() wordt gebruikt in een vergelijking, *moet* CHR(0) aan de linker kant staan in de vergelijking. Wanneer tekenreeksen worden vergeleken, wordt de rechter kant eerst gelezen. Als CHR(0) aan de rechter kant staat, leest dBASE IV niet verder omdat CHR(0) een nultekenreeks is. De logische waarde waar (.T.) wordt weergegeven.

```
. mem = "abc"
abc
. ? mem = CHR(0)
.T.
. ? CHR(0) = mem
.F.
```

Zie ook

???, ASC()

CMONTH()

Met de functie CMONTH() kunt u de naam van de maand in een datumuitdrukking opvragen.

Syntaxis

CMONTH(<Duitdr>)

Gebruik

De datumuitdrukking kan zijn: een geheugenvariabele of een veld, of een functie die een datum oplevert.

Voorbeelden

Als de systeemdatum 15-05-88 is:

```
. ? CMONTH(DATE())
mei
```

In het volgende voorbeeld wordt berekend welke maand het is over 76 dagen. De naam van de maand wordt eerst opgeslagen in een geheugenvariabele:

```
. Mmaand = CMONTH(DATE()+76)
juli

. ? "Wij zouden u graag bezoeken in " + Mmaand
Wij zouden u graag bezoeken in juli
```

Zie ook

DATE(), MONTH()

COL()

Met de functie COL() kunt u opvragen in welke kolom de cursor op dat moment staat.

Syntaxis

COL()

Gebruik

Met behulp van de functie COL() kunt u het beeldscherm relatief adresseren. Met deze functie kunt u de cursor besturen in een programma. Hoewel de cursorpositie ook met de speciale operator $ kan worden opgegeven bij de commando's @...SAY en @...GET, kan dat veel sneller met de functie COL().

Bijvoorbeeld:

@ 1, COL() + 5 is hetzelfde als @ 1, $ + 5.

In beide gevallen wordt de cursor vijf kolommen naar rechts verplaatst.

Voorbeelden

Met COL() kunt u de actieve positie van de cursor op het scherm opvragen. Wanneer u bijvoorbeeld een lus wilt beëindigen wanneer de cursor in de zeventigste kolom staat, kunt u de volgende regels in het programma opnemen:

```
DO WHILE COL() < 70
        @ 5,COL() + 5 SAY 'X'
ENDDO
```

Als u de regel "Dit is relatief adresseren" wilt laten weergeven op rij 7, kolom 10:

```
@ 7, 10 SAY "Dit is "
@ 7, COL() SAY "relatief adresseren"
```

Zie ook

@, PCOL(), PROW(), ROW()

COMPLETED()

Met de functie COMPLETED() kunt u controleren of een transactie is afgerond. Een transactie is afgerond als na het commando BEGIN TRANSACTION het commando END TRANSACTION is gegeven.

Syntaxis

COMPLETED()

Gebruik

Met de functie COMPLETED() kunt u het verloop van een transactie bijhouden vanaf de commandostip of vanuit een programma. De standaardinstelling van deze functie is waar (.T.). Als u het commando BEGIN TRANSACTION geeft, wordt COMPLETED() ingesteld op onwaar (.F.). Wanneer u het commando END TRANSACTION geeft, wordt COMPLETED() weer ingesteld op waar (.T.). Wanneer u ROLLBACK gebruikt voordat u het commando END TRANSACTION geeft, blijft COMPLETED() ingesteld op onwaar (.F.).

Voorbeeld

U kunt op de volgende manier in een programma controleren of een transactie is beëindigd:

```
BEGIN TRANSACTION
   .
   .
   .
END TRANSACTION
IF .NOT. COMPLETED()
   IF .NOT. ROLLBACK()
      ? "De transactie is niet voltooid en de"
      ? "wijzigingen kunnen niet worden hersteld."
      RETURN
   ENDIF
ENDIF
```

Zie ook

BEGIN/END TRANSACTION, CHANGE(), CONVERT, LKSYS(), ROLLBACK, ROLLBACK()

COS()

Met de cosinusfunctie COS() kunt u de cosinus van een hoek in radialen berekenen.

Syntaxis

COS(<Nuitdr>)

Gebruik

De variabele (<Nuitdr>) is een numerieke uitdrukking die de hoek weergeeft in radialen. Er worden geen grenzen gesteld aan deze numerieke uitdrukking.

U kunt het aantal decimalen en de nauwkeurigheid instellen met het commando SET DECIMALS.

Voorbeelden

```
. ?COS(.7854)
            0,7071
```

U kunt de waarde ook toekennen aan een geheugenvariabele:

```
. Mvar =.7071
            0,7071
. Mcos = COS(Mvar)
            0,7602
```

Zie ook

ACOS(), ASIN(), ATAN(), ATN2(), DTOR(), RTOD(), SET DECIMALS, SIN(), TAN()

CTOD()

Met de functie CTOD() kunt u een datum die is ingevoerd en opgeslagen als een tekenreeks, omzetten in een variabele van het type datum.

Syntaxis

CTOD(<Tuitdr>)
of
{dd-mm-jj}

Standaardinstelling

De indeling van de tekenreeks is gewoonlijk dd-mm-jj, maar deze instelling kan worden gewijzigd met de commando's SET DATE en SET CENTURY.

Gebruik

De tekenuitdrukking die wordt opgegeven bij CTOD() moet liggen tussen "01-01-0100" en "31-12-9999". Als u twee cijfers opgeeft voor het jaar, wordt aangenomen dat de datum in de twintigste eeuw valt.

U kunt ook met de accolades {dd-mm-jj} een tekenreeks omzetten in een datum.

Voorbeelden

In het volgende voorbeeld ziet u hoe u een tekenvariabele kunt omzetten in een datumvariabele.

```
. STORE "01-01-80" TO testdatum
01-01-80
. ? TYPE testdatum
C
. STORE CTOD(testdatum) TO nieuwdatum
01-01-80
. ? TYPE("nieuwdatum")
D
```

Bij de alternatieve syntaxis voor de functie CTOD() wordt het datumargument tussen accolades geplaatst.

```
. schrikdat = {29-02-88}
02-29-88
. SET CENTURY ON
. ? schrikdat
02-29-1988
```

Zie ook

DATE(), DTOC(), DTOS(), SET CENTURY, SET DATE

DATE()

Met de functie DATE() kunt u de systeemdatum opvragen. De indeling is dd-mm-jj, tenzij de indeling is gewijzigd met het commando SET CENTURY, SET DATE of SET MARK.

Syntaxis

DATE()

Gebruik

De systeemdatum moet zijn ingesteld voordat u dBASE IV start.

Voorbeelden

U kunt de systeemdatum als volgt weergeven:

```
. ? DATE()
04-01-88
```

U kunt de systeemdatum ook opslaan in een geheugenvariabele:

```
. STORE DATE() TO Mdatum
05-04-88
```

of:

```
. Mdatum = DATE()
29-02-88
```

Om het aantal dagen tussen twee datums te berekenen typt u:

```
. ? DATE()
08-15-90
. Thirdq = {07-01-90}
07-01-90
. ? "Dagen" + STR(DATE() - Thirdq,5,0)
Dagen                45
```

Zie ook

SET CENTURY, SET DATE, SET MARK

DAY()

Met de functie DAY() kunt u de numerieke waarde opvragen van de dag in een datumuitdrukking.

Syntaxis

DAY(<Duitdr>)

Gebruik

De datumuitdrukking is een veld of een geheugenvariabele, of een functie die een datum oplevert.

Voorbeelden

Als de systeemdatum 15-05-88 is:

```
. DAY(DATE())
15
```

U kunt de dag van de systeemdatum ook opslaan in een geheugenvariabele:

```
. STORE DAY(DATE()) TO Mdag
                    15,00
. ? Mdag
                    15,00

. ? TYPE("Mdag")
N
```

Zie ook

CDOW(), DATE(), DOW()

DBF()

Met de functie DBF() kunt u de naam opvragen van het actieve database-bestand in het geselecteerde werkgebied.

Syntaxis

DBF([<alias>])

Standaardinstelling

De standaardinstelling van DBF() is het actieve werkgebied, tenzij er een alias is opgegeven. Als er geen database-bestand is geopend in het opgegeven werkgebied, levert DBF() een nultekenreeks op.

Gebruik

Deze functie is handig als u een programma schrijft dat moet werken met een bestand dat actief is in een ander werkgebied.

Voorbeelden

Op de volgende manier kunt u in de interactieve modus bepalen welk bestand actief is:

```
. USE Afnemers
. ? DBF()
C:AFNEMERS.DBF
```

In het volgende voorbeeld worden de database-bestanden in alle werkgebieden gesloten, maar wordt het actieve database-bestand weer geopend:

```
. bestandsnaam = DBF()
. CLOSE ALL
. USE (bestandsnaam)
```

Zie ook

ALIAS(), FIELD(), KEY(), MDX(), NDX(), SET FULLPATH, TAG()

DELETED()

Met de functie DELETED() kunt u controleren of een record een wismarkering heeft. Als het actieve record in het opgegeven werkgebied een wismarkering heeft, levert dat de logische waarde waar (.T.) op. Als het record geen wismarkering heeft, levert dat de logische waarde onwaar (.F.) op.

Syntaxis

DELETED([<alias>])

Speciaal geval

Als er geen database-bestand actief is, wordt de logische waarde onwaar (.F.) weergegeven. Als u geen alias opgeeft, wordt het record in het actieve werkgebied gecontroleerd.

Voorbeeld

U kunt als volgt bepalen of het actieve record een wismarkering heeft:

```
. USE Afnemers
. ? DELETED()
.F.
. DELETE
1 record gewist.
. ? DELETED()
.T.
```

Zie ook

PACK, RECALL, SET DELETED

DIFFERENCE()

Met de functie DIFFERENCE () kunt u het verschil tussen twee tekenreeksen bepalen.

Syntaxis

DIFFERENCE(<Tuitdr>, <Tuitdr>)

Gebruik

Met de functie DIFFERENCE() worden de twee opgegeven tekenreeksen omgezet in SOUNDEX()-codes en wordt het verschil tussen de twee uitdrukkingen bepaald. Het resultaat is een geheel getal tussen 0 en 4. Wanneer de twee codes sterk op elkaar lijken, wordt de waarde 4 teruggezonden. Wanneer de twee codes geen enkele letter gemeen hebben, wordt de waarde 0 teruggezonden. Wanneer de tekenreeksen één letter gemeen hebben, wordt de waarde 1 teruggezonden.

De twee uitdrukkingen die worden vergeleken moeten tekenuitdrukkingen zijn. U kunt ook de namen opgeven van variabelen die zijn gedefinieerd.

Voorbeeld

In het volgende voorbeeld wordt gezocht naar namen met ongeveer dezelfde SOUNDEX()-codes:

```
. USE Afnemers
. LIST Voornaam
Recordnr.      Voornaam
        1      Fred
        2      Sandra
        3      Rinus
        4      Kim
        5      Ger
        6      Gemma
        7      Floor
        8      Rien
. nieuwnaam = "Frederik"
Frederik
. LIST Voornaam FOR DIFFERENCE(Voornaam, nieuwnaam) > 1
Recordnr.      Voornaam
        1      Fred
        2      Sandra
        7      Floor
. LIST Voornaam FOR DIFFERENCE(Voornaam, nieuwnaam) > 2
Record#        Voornaam
        4      Fred
```

Zie ook

SOUNDEX()

DISKSPACE()

Met de functie DISKSPACE() kunt u opvragen hoeveel ruimte, uitgedrukt in bytes, er nog beschikbaar is op het standaardstation.

Syntaxis

DISKSPACE()

Richtlijnen voor het programmeren

Wanneer een toepassingsprogramma automatisch een reservekopie van een database-bestand maakt, kunt u met de functies DISKSPACE(), RECCOUNT() en RECSIZE() controleren of er op het station nog genoeg ruimte is voor het reservebestand. Zie het voorbeeld bij de functie RECSIZE().

Voorbeeld

```
SET TALK OFF
USE Afnemers
Mbstgroot = (RECCOUNT() * RECSIZE()) + 2000
IF DISKSPACE() >Mbstgroot
    Copy TO A:Backup
ELSE
    @ 10,16 SAY "Niet genoeg ruimte voor reservekopie."
ENDIF
SET TALK ON
```

Zie ook

RECCOUNT(), RECSIZE()

DMY()

Met de functie DMY() kunt u elke geldige datumuitdrukking omzetten in een datum met de indeling dag-maand-jaar.

Syntaxis

DMY(<Duitdr>)

Gebruik

Met de functie DMY() wordt een datum omgezet in de volgende indeling:

DD maand JJ

De dag wordt weergegeven met één of twee cijfers, zonder voorloopnullen. De naam van de maand wordt voluit weergegeven. Van het jaar worden alleen de laatste twee cijfers weergegeven.

Als SET CENTURY ON is ingesteld, is de indeling:

DD maand JJJJ

De datumuitdrukking kan een geheugenvariabele of een veld zijn, of een functie die een datum oplevert.

Voorbeeld

In het volgende voorbeeld wordt een datum weergeven met de functie DMY():

```
. schrikdat = {29-02-88}
29-02-88
. ? DMY(schrikdat)
29 februari 88
```

Zie ook

CDOW(), CMONTH(), DATE(), DOW(), MDY(), MONTH(), SET CENTURY, SET DATE, YEAR()

DOW()

Met de functie DOW() kunt u van een datumuitdrukking de dag van de week opvragen. De dag wordt weergegeven als een cijfer, waarbij zondag dag 1 is.

Syntaxis

DOW(<Duitdr>)

Gebruik

De datumuitdrukking kan een geheugenvariabele of een veld zijn, of een functie die een datum oplevert.

Voorbeelden

Stel dat 13-05-88 de systeemdatum is:

```
. ? DOW(DATE())
                    6.00
```

U kunt de dag van de week opslaan in een geheugenvariabele:

```
. DOW(DATE()) = Werkdag
                    6,00
. ? Werkdag
                    6,00
```

In het volgende voorbeeld wordt berekend welke dag het is over 197 dagen:

```
. ? DOW(DATE() + 197)
                    7,00
```

Zie ook

CDOW(), DATE(), DAY()

DTOC()

Met de functie DTOC() kunt u een datumuitdrukking omzetten in een tekenreeks.

Syntaxis

DTOC(<Duitdr>)

Gebruik

Met de functie DTOC() kunt u een datum opslaan in een geheugenvariabele van het type teken, zodat u een datum kunt vergelijken met een teken.

Voorbeelden

U kunt de systeemdatum als volgt opslaan in een geheugenvariabele van het type teken (gesteld dat de systeemdatum 13-05-88 is):

```
. STORE DTOC(DATE()) TO testdatum
13-05-88
. ? TYPE("testdatum")
C
```

U kunt een datumvariabele ook koppelen aan een tekenreeks, zodat de datum als tekst kan worden uitgedrukt:

```
. STORE DATE() TO testdatum
13-05-88
. ? TYPE("testdatum")
D
. STORE "De datum is: "+DTOC(testdatum) TO text
De datum is: 13-05-88
```

Zie ook

CTOD(), DATE(), DTOS(), SET CENTURY, SET DATE

DTOR()

Met de functie DTOR() kunt u graden omzetten in radialen.

Syntaxis

DTOR(<Nuitdr>)

Gebruik

<Nuitdr> is de hoek gemeten in graden. De uitkomst van de functie DTOR() is de hoek gemeten in radialen.

U kunt geen minuten en seconden opgeven bij <Nuitdr>. Zet minuten en seconden eerst om in decimalen.

Voorbeelden

```
. ? DTOR(180)
                    3,14
```

In het volgende voorbeeld wordt een hoek van 60 graden 30 minuten en 15 seconden omgezet in radialen:

```
.DTOR(60,525)
                    1,056
```

U kunt ook een variabele omzetten:

```
. Variabele1 = 12
                    12,00
. Variabele2 = DTOR(Variabele1)
                    0,21
```

Zie ook

ACOS(), ATAN(), ATN2(), COS(), RTOD(), SET DECIMALS, SIN()

DTOS()

Met de functie DTOS() kunt u een datumuitdrukking omzetten in een tekenreeks.

Syntaxis

DTOS(<Duitdr>)

Gebruik

Met de functie DTOS() kunt u indexeren op een datumuitdrukking gecombineerd met een tekenuitdrukking. De variabele <Duitdr> is een datumuitdrukking. Met deze functie wordt de opgegeven datum omgezet in een tekenreeks met de indeling JJJJMMDD, ongeacht de instelling van SET CENTURY of DATE SETTING. Hiermee wordt bereikt dat de datum op de juiste manier wordt geïndexeerd als de datum wordt gekoppeld met tekenreeksen in een indexuitdrukking.

Voorbeeld

```
.USE Transact
. INDEX ON DTOS(Dat_trans) + Bestel_nr TO Tr_datum
100% geïndexeerd              12 records geïndexeerd

. LIST NEXT 6 Dat_trans, Bestel_nr
Recordnr.   Dat_trans     Bestel_nr
        1   02-02-87      87-105
        2   10-02-87      87-106
        3   12-02-87      87-107
        4   23-02-87      87-108
        5   09-03-87      87-109
        6   09-03-87      87-110
```

Zie ook

CTOD(), DATE(), DTOC(), SET CENTURY, SET DATE

EOF()

De functie EOF() geeft het einde van een bestand aan. Wanneer het laatste logische record van het opgegeven bestand is gepasseerd, wordt de logische waarde waar (.T.) weergegeven.

Syntaxis

EOF([<alias>])

Gebruik

Wanneer EOF() waar (.T.) is, is RECNO() gelijk aan RECCOUNT() + 1. Daarnaast geven bepaalde commando's (bijvoorbeeld SKIP) de foutmelding **Einde van bestand bereikt**. Als u het commando SKIP geeft terwijl de recordwijzer op het laatste record van het bestand staat, geeft RECNO() een waarde aan die één hoger is dan het aantal records in het bestand, en levert EOF() waar (.T.) op.

Speciale gevallen

Als u geen alias opgeeft, wordt het actieve database-bestand in het actieve werkgebied gecontroleerd.

Als er geen database-bestand is geopend in het opgegeven werkgebied, is de logische waarde onwaar (.F.) het resultaat.

Voorbeeld

U kunt op de volgende manier controleren of de recordwijzer het laatste logische record van het database-bestand is gepasseerd:

```
.USE Afnemers
. GO BOTTOM
. ? EOF()
.F.
. SKIP
. ? EOF()
.T.
```

Zie ook

BOF(), ERROR(), FIND, FOUND(), LOCATE, RECCOUNT(), RECNO(), SEEK, SEEK()

ERROR()

Met de functie ERROR() kunt u het nummer opvragen van de foutmelding van dBASE IV die werd gegeven naar aanleiding van een foutsituatie.

Syntaxis

ERROR()

Gebruik

Met de functie ERROR() kunt u het foutnummer opvragen van de fout die werd gesignaleerd door het commando ON ERROR. Bijlage A van dit boek bevat een lijst met foutmeldingen en de bijbehorende foutnummers met, indien mogelijk, suggesties voor het verhelpen van de fouten.

OPMERKING
Er wordt alleen een foutnummer weergegeven als een ON ERROR-commando actief is. Met de commando's RETURN en RETRY kunt u het foutnummer en de foutmelding wissen.

Voorbeeld

In een programma waarin de gebruiker wordt gevraagd welk PFS-bestand moet worden geïmporteerd, wordt een routine opgenomen die controleert of de foutmelding **Geen geldig PFS-bestand** wordt gegeven. Het foutnummer van deze melding is 140:

```
* Hoofd.PRG
ON ERROR DO Fout_proc
ACCEPT "Geef de naam van het PFS-bestand op: " TO bestnaam
IMPORT FROM &(bestnaam) TYPE PFS
* PROCEDURE Fout_proc
ON ERROR
DO CASE
    CASE ERROR() = 140
        ? "(bestnaam) is geen PFS-bestand."
        *
        *
        *
ENDCASE
RETURN
```

Zie ook

CERROR, DEBUG, LINENO(), MESSAGE(), ON ERROR, PROGRAM(), RETRY, RETURN

EXP()

Met de functie EXP() kunt u de constante e verheffen tot de macht <Nuitdr>.

Syntaxis

EXP(<Nuitdr>)

Gebruik

In de vergelijking $y = e^x$ is <Nuitdr> de waarde van x. Voor elke exponent x van de basis e wordt de waarde van y berekend. Het getal dat wordt weergegeven is een reëel getal. Met SET DECIMALS kunt u de nauwkeurigheid van het weergegeven antwoord opgeven.

Voorbeelden

Wanneer een getal wordt verheven tot de macht 1, is de uitkomst gelijk aan het getal. Wanneer e tot de macht 1 wordt verheven, is de uitkomst de waarde van e.

```
. ? EXP(1.000)
                    2,72
```

In het volgende voorbeeld wordt een logaritmische berekening uitgevoerd:

```
. x = log(25) + log(25)
                    6,44
. ? EXP(x)
                    625,00
```

Zie ook

LOG(), SET DECIMALS, SET PRECISION

FIELD()

Met de functie FIELD() kunt u het veldnummer opvragen van een bepaald veld in de bestandsstructuur van het actieve database-bestand.

Syntaxis

FIELD(<Nuitdr>[,<alias>])

Gebruik

Als de optionele aliasnaam niet wordt opgegeven, wordt uitgegaan van het actieve werkgebied.

De veldnaam die wordt teruggezonden staat altijd in hoofdletters. Als de numerieke uitdrukking kleiner is dan 1 of groter dan 255, levert dat een nultekenreeks op. Dat gebeurt ook als de numerieke uitdrukking verwijst naar een ongebruikt veld. Wanneer de bestandsstructuur bijvoorbeeld 28 veldnamen bevat en u een veldnummer boven de 28 opgeeft, levert dat een nultekenreeks op.

Omdat bij de functie FIELD() de velden met nummers worden geadresseerd, kunt u velden behandelen als reeksen.

Voorbeeld

Gebruik het database-bestand Afnemers.

```
. USE Afnemers
. ? FIELD(3)
ACHTERNAAM
. getal = 2
                    2.00
. Mveld = FIELD(getal)
KLANTNAAM
```

Zie ook

DBF(), LUPDATE(), RECCOUNT(), RECSIZE(), VARREAD()

FILE()

Met de functie FILE() kunt u controleren of een bepaald bestand op een bepaald station voorkomt. Als het bestand wordt gevonden, levert dat de logische waarde waar (.T.) op.

Syntaxis

FILE(<Tuitdr>)

Gebruik

Bij de functie FILE() moet u zowel de bestandsnaam als de extensie opgeven. Als het bestand niet op het standaardstation of in de standaard bestandsindex staat, moet u ook het station en de bestandsindex opgeven.

Bij de functie FILE() wordt geen onderscheid gemaakt tussen hoofdletters en kleine letters.

Bij de functie FILE() wordt kort geprobeerd een bestand te openen om te controleren of het bestaat.

Voorbeelden

Als C het standaardstation is en Afnemers.dbf op station A staat, wordt het bestand niet gevonden als u het station niet opgeeft:

```
. ? FILE("Afnemers.dbf")
.F.
```

Zonder extensie wordt het bestand ook niet gevonden:

```
. ? FILE("A:Afnemers")
.F.
```

U geeft dus de volledige bestandsaanduiding op:

```
. ? FILE("A:Afnemers.dbf")
.T.
```

Zie ook

SET DEFAULT, SET DIRECTORY, SET PATH

FIXED()

Met de functie FIXED() kunt u lange reële getallen met een drijvende komma omzetten in binair gecodeerde decimale getallen.

Syntaxis

FIXED(<Nuitdr>)

Gebruik

Met de functie FIXED() kunt u getallen van het type F (IEEE lang, reëel getal met drijvende komma) omzetten in getallen van het type N (binair gecodeerde decimaal). Bij het omzetten kan iets van de nauwkeurigheid verloren gaan. De wetenschappelijke notatie wordt voor beide typen getallen ondersteund.

Als in een uitdrukking beide typen getallen voorkomen, worden alle getallen omgezet in type F.

Het bereik loopt van 10308 tot en met 10-308

Zie ook

FLOAT(), SET PRECISION

FKLABEL()

Met de functie FKLABEL() kunt u de naam opvragen die aan de opgegeven functietoets is toegewezen.

Syntaxis

FKLABEL(<Nuitdr>)

Gebruik

U kunt in het totaal 28 toetsen programmeren: 9 functietoetsen en 19 combinaties van functietoetsen met **Shift** of **Ctrl**. Met het schermgrote commando SET kunt u een lijst van alle programmeerbare toetsen oproepen. U kunt deze toetsen programmeren vanuit de menu's van het schermgrote commando SET of vanaf de commandostip met het commando SET FUNCTION. U kunt de functietoetsen ook programmeren in het bestand Config.db (raadpleeg hoofdstuk 6, *dBASE IV aanpassen*.)

Met de functie FKLABEL() kunt u de label opvragen van de functietoets die wordt aangeduid met <Nuitdr>. 1 tot en met 28 zijn geldige numerieke uitdrukkingen.

De functietoets **F1** is gereserveerd voor de hulpfunctie en kan niet worden geprogrammeerd. De nummering van de functietoetsen begint bij functietoets **F2 (F2 = 1)**.

Voorbeeld

Om de FKLABEL() te bepalen die overeenkomt met een bepaalde functietoets, typt u het volgende vanaf de stip:

```
. ? FKLABEL(9)
F10
```

Zie ook

FKMAX(), SET, SET FUNCTION

FKMAX()

Met de functie FKMAX() kunt u opvragen hoeveel programmeerbare functietoetsen uw toetsenbord heeft.

Syntaxis

FKMAX()

Gebruik

Met de functie FKMAX() kunt u bepalen hoeveel programmeerbare toetsen worden ondersteund door dBASE IV.

De functietoets **F1** is gereserveerd voor de hulpfunctie van dBASE IV en **Shift-F10** is gereserveerd voor het macromenu. Deze toetsen kunnen dus niet worden geprogrammeerd. De functietoetsen **F11** en **F12** van een uitgebreid toetsenbord kunnen ook niet worden geprogrammeerd.

Voorbeeld

Hoeveel toetsen kunnen er worden geprogrammeerd?

```
.? FKMAX()
28
```

Dit is het totaal van **F2** tot en met **F10** (9 toetsen), **Shift-F1** tot en met **Shift-F9** (9 toetsen) en **Ctrl-F1** tot en met **Ctrl-F10** (10 toetsen).

Zie ook

FKLABEL(), SET, SET FUNCTION

FLOAT()

Met de functie FLOAT() kunt u binair gecodeerde decimale getallen (type N) omzetten in lange, reële getallen met een drijvende komma (type F).

Syntaxis

FLOAT(<Nuitdr>)

Gebruik

<Nuitdr> kan elke geldige uitdrukking van het type N zijn. De uitdrukking wordt omgezet in een getal van het type F.

Het bereik van <Nuitdr> is gebaseerd op de instelling van SET PRECISION TO <Nuitdr>. Het hoogste toegestane getal is 0,9xE+308. Het kleinste toegestane getal 0,1xE-307. Voor het bereik bij getallen van het type F gelden dezelfde regels.

Een uitdrukking die een getal van het type F bevat levert als resultaat ook een getal van het type F op.

Zie ook

FIXED()

FLOCK()

Met de functie FLOCK() kunt u een database-bestand vergrendelen vanaf de commandostip of vanuit een programma.

Syntaxis

FLOCK([alias])

Gebruik

Met de functie FLOCK() kunt u voorkomen dat meerdere gebruikers tegelijkertijd hetzelfde bestand bijwerken. Met de functie FLOCK() kunt u alle records in een bestand vergrendelen voor commando's waarbij het gehele bestand wordt bewerkt. Als u slechts één record bewerkt, kunt u met de functies RLOCK() en LOCK() alleen dat record vergrendelen. Daardoor blijven de overige records toegankelijk voor de andere gebruikers in het netwerk.

Deze functie is een aanvulling op de automatische vergrendeling van dBASE IV die wordt geactiveerd wanneer u een commando geeft dat het gehele database-bestand bewerkt. Raadpleeg het gedeelte over het commando SET LOCK in hoofdstuk 3 van dit boek voor een lijst met alle commando's van dBASE IV die een database-bestand of record automatisch vergrendelen.

Een vergrendeld bestand kan niet worden gewijzigd door een andere gebruiker totdat het bestand wordt ontgrendeld. Andere gebruikers kunnen het bestand echter wel bekijken.

Als u de optionele aliasnaam opgeeft bij FLOCK(), wordt geprobeerd het bestand in het opgegeven werkgebied te vergrendelen. Als u geen alias opgeeft, wordt het bestand in het actieve werkgebied vergrendeld. De alias die u opgeeft moet dezelfde naam hebben als het sleutelwoord ALIAS of de standaard bestandsnaam.

Als het bestand met succes wordt vergrendeld, wordt het bestand voor exclusief gebruik geopend en wordt de logische waarde waar (.T.) weergegeven. Het bestand blijft vergrendeld tot u:

- het bestand ontgrendelt met het commando UNLOCK
- het bestand sluit
- stopt met dBASE IV

Als een bestand al is vergrendeld door een andere gebruiker, wordt de logische waarde onwaar (.F.) weergegeven als u dat bestand probeert te vergrendelen. Als dBASE IV individueel wordt gebruikt, wordt altijd de logische waarde waar (.T.) weergegeven door FLOCK().

Als u een bestand vergrendelt terwijl het actief is gekoppeld aan andere bestanden, worden alle bestanden in de relatie automatisch vergrendeld. Als een van de bestanden wordt ontgrendeld, worden alle gekoppelde bestanden ook automatisch ontgrendeld.

Voorbeeld

In het volgende deel van een programma wordt met de functie FLOCK() een database-bestand vergrendeld. Als het bestand na 250 pogingen nog niet is vergrendeld, wordt de procedure Blok_tijd uitgevoerd:

```
SET REPROCESS TO 250
IF .NOT. FLOCK()
    DO Blok_tijd
    RETURN
ENDIF
```

Zie ook

ACCESS(), RLOCK( )/LOCK( ), SET LOCK, SET REPROCESS, UNLOCK

FLOOR()

Met de functie FLOOR() kunt u het grootste gehele getal bepalen dat kleiner dan of gelijk is aan de opgegeven numeriek uitdrukking.

Syntaxis

FLOOR(<Nuitdr>)

Gebruik

Met de functie FLOOR() kunt u het grootste gehele getal bepalen dat kleiner dan of gelijk is aan de opgegeven waarde. De waarde die wordt teruggezonden is van hetzelfde gegevenstype als de numerieke uitdrukking <Nuitdr>.

Voorbeeld

```
. eerste = 121
                    121.00
. ? FLOOR(eerste / 10)
                    12.00
```

Zie ook

CEILING(), INT(), MOD(), ROUND()

FOUND()

Met de functie FOUND() kunt u controleren of een van de voorgaande zoekcommando's FIND, LOCATE, SEEK of CONTINUE succesvol was. Als het commando in het geselecteerde werkgebied succesvol was, levert dat de logische waarde waar (.T.) op.

Syntaxis

FOUND([<alias>])

Gebruik

Met de functie FOUND() kunt u de loop van het programma beïnvloeden aan de hand van het resultaat van een zoekcommando. Er is een aparte FOUND() voor ieder werkgebied. Als u geen alias opgeeft, wordt FOUND() uitgevoerd op het actieve werkgebied.

Als bestanden zijn gekoppeld met het commando SET RELATION TO, worden de gekoppelde database-bestanden ook doorzocht als u het commando FIND, LOCATE, SEEK of CONTINUE geeft voor het actieve werkgebied.

Als de recordwijzer wordt verplaatst, maar niet met het commando FIND, LOCATE, SEEK of CONTINUE, levert de functie FOUND() de logische waarde onwaar (.F.) op.

Voorbeeld

In het volgende voorbeeld wordt gecontroleerd of een ongeïndexeerd bestand met succes is doorzocht:

```
.USE Afnemers
. LOCATE FOR Achternaam = "Peters"
Record = 4
. ? FOUND()
.T.
. CONTINUE
Einde van LOCATE-bereik
.? FOUND()
.F.
```

Zie ook

CONTINUE, EOF(), FIND, LOCATE, LOOKUP(), SEEK, SEEK(), SET NEAR, SET RELATION

FV()

Met de functie FV() kunt u de toekomstige waarde berekenen van regelmatige stortingen van een vast bedrag in een investering met een vaste rentevoet over een bepaald aantal termijnen.

Syntaxis

FV(<bedrag>,<rente>,<termijn>)

Gebruik

<bedrag> is een numerieke uitdrukking die zowel positief als negatief kan zijn.

<rente> is de rentevoet. Dit is altijd een positief getal. Het is de rentevoet over de opgegeven periode. Als u een jaarlijkse rentevoet hanteert en u het bedrag per maand wilt berekenen, geeft u bij <rente> de rentevoet gedeeld door 12 op.

<termijn> is een numerieke uitdrukking die het aantal stortingen weergeeft. Een termijn is de vaste tijd tussen elke storting.

Het resultaat van FV() is een getal met een vast decimaalteken. Het getal is het totaal van de gestorte bedragen en de samengestelde rente.

Voorbeeld

```
. INPUT "Geef storting op: " TO storting
Geef storting op: 1,00
. INPUT "Geef rente op: " TO rente
Geef rente op: ,02
. INPUT Geef aantal stortingen op: TO termijn
Geef aantal stortingen op: 24
. ? FV(storting, rente, termijn)
                    30,42
```

Zie ook

CALCULATE, PV()

GETENV()

Met de functie GETENV() kunt u de inhoud van een omgevingsvariabele van DOS opvragen, bijvoorbeeld PATH of COMSPEC. De naam van de systeemvariabele kan worden ingevoerd als een tekenreeks, een tekenvariabele of een combinatie hiervan.

Syntaxis

GETENV(<Tuitdr>)

Gebruik

Met GETENV() kunt u de instelling van de omgevingscommando's van DOS controleren. Als de parameter van het omgevingscommando niet wordt gevonden, wordt een nultekenreeks teruggezonden.

De variabelen van de besturingssysteemomgeving zijn commando's als PROMPT, PATH en COMSPEC. Raadpleeg het boek bij het DOS-besturingssysteem voor een lijst met de omgevingscommando's. Deze commando's zijn meestal ingesteld in het bestand Autoexec.bat.

Raadpleeg het gedeelte over het commando SHELL in het DOS-boek als u meer ruimte wilt reserveren voor de omgeving. Als u veel DOS-configuratiecommando's of lange zoekpaden opgeeft, kan het zijn dat er niet genoeg ruimte is voor de omgeving.

Voorbeeld

Aangenomen dat C het standaardstation is en dat u meerdere bestandsindexen hebt opgegeven bij het commando PATH, kunt u als volgt de instelling van "PATH" controleren:

```
. ? GETENV("path")
C:\;C:\DOS;C:\DBASE...
```

Alle andere bestandsindexen die u hebt opgegeven bij PATH, worden ook weergegeven.

Zie ook

OS()

IIF()

IIF() staat voor *immediate IF* en is een verkorte notatie van de constructie IF...ENDIF.

Syntaxis

IIF(<voorwaarde>,<uitdr1>,<uitdr2>)

Gebruik

U kunt de functie IIF() gebruiken vanaf de commandostip of in een programma. Als de logische uitdrukking waar (.T.) is, wordt de uitkomst van de eerste uitdrukking weergegeven. Als de uitkomst onwaar (.F.) is, wordt de uitkomst van de tweede uitdrukking weergegeven.

<uitdr1> en <uitdr2> kunnen uitdrukkingen van het type teken, numeriek, logisch of datum zijn. Beide uitdrukkingen moeten van hetzelfde type zijn. Met de functie IFF() worden <uitdr1> en <uitdr2> niet geëvalueerd. De functie levert de uitdrukkingen op wanneer de voorwaarde waar of onwaar is.

U kunt de functie IIF() geven vanaf de commandostip, of met commando's als CREATE/MODIFY REPORT en CREATE/MODIFY LABEL. Met de functie IIF() kunt u een veld of een waarde in een uitdrukking vervangen door één van de twee waarden, afhankelijk van de voorwaarde. Dit is nuttig bij het maken van een query, rapport, etiket of index.

Voorbeelden

In het volgende programma kunt u de constructie IF...ENDIF vervangen door één regel met de functie IFF(). Het eerste voorbeeld is de constructie IF...ENDIF:

```
IF Geslacht = "V"
    Mnaam = "Mevr. " + Achternaam
ELSE
    Mnaam = "Dhr. " + Achternaam
ENDIF
```

Het voorgaande voorbeeld kan worden verkort tot slechts één regel in het programma:

```
Mname = IIF(Geslacht = "V", "Mevr. ", "Dhr. ") + Achternaam
```

Met deze constructie kunt u de juiste titel voor de naam afdrukken bij het maken van etiketten.

Als een database-bestand de velden Begindatum en Einddatum bevat, kunt u Einddatum met 30 dagen verhogen als de termijn is verlopen:

In de interactieve modus:

```
. REPLACE ALL Eind_dat WITH IIF(Einddatum >DATE(), Einddatum, Einddatum + 30)
```

Als u al een waarde hebt toegekend aan de logische geheugenvariabele Is_fout, kunt u met de volgende routine de variabele omzetten in een getal en dat getal opslaan in de numerieke variabele Foutcode. Als Is-fout waar (.T.) is, krijgt de nieuwe geheugenvariabele de waarde 1, anders de waarde 0.

```
foutcode = IIF(is_fout, 1, 0)
```

INKEY()

Met de functie INKEY() kunt u een geheel getal opvragen dat de toets weergeeft die het laatst is ingedrukt. De uitvoering van een programma wordt met deze functie niet onderbroken.

Syntaxis

INKEY([n])

Gebruik

Deze functie wordt in programma's vaak gebruikt als voorwaarde voor een sprongopdracht. De functie INKEY() maakt het gebruik van functietoetsen en cursortoetsen in een programma van dBASE IV mogelijk.

Het resultaat van de functie INKEY() is een geheel getal tussen 0 en 255, dat overeenkomt met de decimale ASCII-waarde van het teken in de typebuffer. Voor speciale toetsen, bijvoorbeeld de cursortoetsen en **Ctrl-Home**, wordt de waarde van een combinatie van een toets met **Ctrl** weergegeven. Voor combinaties van functietoetsen met **Shift** of **Ctrl** wordt een negatieve waarde weergegeven.

Wanneer er meerder tekens in de typebuffer staan, zal INKEY() de waarde van het eerste teken in de typebuffer weergeven, waarna het teken uit de buffer wordt verwijderd.

Met het optionele numerieke argument wordt het aantal seconden gespecificeerd dat INKEY() wacht tot de een functietoets wordt ingedrukt. Na het verstrijken van die tijd komt u automatisch weer in het programma van waaruit INKEY() is aangeroepen. Het numerieke argument kan ook minder dan een een seconde zijn. Als u INKEY(0) opgeeft, is de wachttijd onbeperkt.

Als er geen toets is ingedrukt, levert INKEY() (zonder parameter) de waarde 0 op.

De decimale codes van alle toetsen en toetscombinaties (behalve de combinaties met **Alt**) worden getoond in tabel 4-1. De toetscombinaties **Ctrl-S** (19) en en ← worden niet geregistreerd door INKEY(), tenzij SET ESCAPE op OFF is ingesteld.

De toetscombinatie **Alt**-functietoets is gereserveerd voor macro's en wordt niet geregistreerd door INKEY().

Tabel 4-2 laat de toetsenbordcodes zien die worden weergegeven voor combinaties van **Alt** met een alfanumerieke toets. De codes voor toetscombinaties met **Alt** zijn hetzelfde voor hoofdletters en kleine letters. De waarden die worden weergegeven zijn decimaal.

Tabel 4-1 Resulterende waarde van INKEY()

Toetsnaam		Decimale waarde
Ctrl-A	**Ctrl**-←	1
Ctrl-B	**End**	2
Ctrl-C	**PgDn**	3
Ctrl-D	→	4
Ctrl-E	↑	5
Ctrl-F	**Ctrl**-→	6
Ctrl-G	**Del**	7
Ctrl-H		8
Ctrl-I	**Tab**	9
Ctrl-J		10
Ctrl-K		11
Ctrl-L		12
Ctrl-M		13
Ctrl-N		14
Ctrl-O		15
Ctrl-P		16
Ctrl-Q		17
Ctrl-R	**PgUp**	18
Ctrl-S	←	19
Ctrl-T		20
Ctrl-U		21
Ctrl-V	**Ins**	22
Ctrl-W	**Ctrl-End**	23
Ctrl-X	↓	24
Ctrl-Y		25
Ctrl-Z	**Home**	26

(wordt vervolgd)

Tabel 4-1 Resulterende waarde van INKEY() (vervolg)

Toetsnaam		Decimale waarde
Esc	Ctrl-[	27
F1	Ctrl-\	28
Ctrl-]	Ctrl-Home	29
Ctrl-^	Ctrl-PgDn	30
Ctrl-PgUp		31
Spatiebalk		32
Spatie terug		127
Tab terug		-400
Ctrl-Spatie-terug		-401
Ctrl-↵		-402
F2		-1
F3		-2
F4		-3
F5		-4
F6		-5
F7		-6
F8		-7
F9		-8
F10		-9
Ctrl-F1		-10
Ctrl-F2		-11
Ctrl-F3		-12
Ctrl-F4		-13
Ctrl-F5		-14
Ctrl-F6		-15
Ctrl-F7		-16

(wordt vervolgd)

Tabel 4-1 Resulterende waarde van INKEY() (vervolg)

Toetsnaam	Decimale waarde
Ctrl-F8	-17
Ctrl-F9	-18
Ctrl-F10	-19
Shift-F1	-20
Shift-F2	-21
Shift-F3	-22
Shift-F4	-23
Shift-F5	-24
Shift-F6	-25
Shift-F7	-26
Shift-F8	-27
Shift-F9	-28
Shift-F10	-29

Tabel 4-2 **Alt**-combinaties van INKEY()

Met letters	Met letters	Met letters	Met cijfers
a - 435	k - 425	u - 415	1 - 451
b - 434	l - 424	v - 414	2 - 450
c - 433	m - 423	v - 414	3 - 449
d - 432	n - 422	w - 413	4 - 448
e - 431	o - 421	x - 412	5 - 447
f - 430	p - 420	y - 411	6 - 446
g - 429	q - 419	z- 410	7 - 445
h - 428	r - 418		8 - 444
i - 427	s - 417		9 - 443
j - 426	t - 416		0 - 452

Voorbeeld

In het volgende programma wordt, afhankelijk van de toets die de gebruiker heeft ingedrukt, naar een bepaalde procedure gesprongen. Met het numerieke argument geeft u op dat het programma drie seconden moet wachten op een toetsaanslag van de gebruiker voordat het programma verder wordt uitgevoerd.

```
DO WHILE .T.
    i = INKEY(3)
    IF i<=0
       i=0
    ENDIF
    DO CASE
            CASE CHR(i) $ "Qq0"
                    RETURN
       CASE CHR(i) $ "Aa1"
                    DO <procedure 1>
       CASE CHR(i) $ "Bb2"
                    DO <procedure 2>
    ENDCASE
ENDDO
```

Zie ook

CHR(), KEYBOARD, LASTKEY(), ON KEY, READ, READKEY(), SET TYPEAHEAD

INT()

Met de functie INT() kunt u een numerieke uitdrukking inkorten tot een geheel getal.

Syntaxis

INT(<Nuitdr>)

Gebruik

Met de functie INT() kunt u bij een numerieke uitdrukking alle cijfers achter het decimaalteken laten vervallen.

Voorbeeld

U kunt het getal 10,23 als volgt omzetten in een geheel getal:

```
. ? INT(10.23)
                    10
. STORE 10.23 TO x
                    10,23
. STORE INT(10.23) TO x
                    10
```

Zie ook

CEILING(), FLOOR(), ROUND()

ISALPHA()

Met de functie ISALPHA() kunt u controleren of de opgegeven tekenuitdrukking begint met een letter. Als de tekenuitdrukking met een letter begint wordt de logische waarde waar (.T.) weergegeven.

Syntaxis

ISALPHA(<Tuitdr>)

Gebruik

De letter kan zowel een hoofdletter als een kleine letter zijn.

Met deze functie kunt u bepalen of een tekenreeks met een letter begint.

Voorbeelden

U kunt als volgt controleren of een tekenreeks met een letter begint:

```
. ? ISALPHA("abc123")
.T.
. ? ISALPHA("ABC123")
.T.
. ? ISALPHA("123abc")
.F.
```

Zie ook

ISLOWER(), ISUPPER(), LOWER(), UPPER()

ISCOLOR()

Met de functie ISCOLOR() kunt u nagaan of een computer kleuren op het beeldscherm kan weergeven.

Syntaxis

ISCOLOR()

Gebruik

Met de functie ISCOLOR() kunnen programmeurs toepassingen ontwerpen voor zowel kleurenbeeldschermen als monochrome beeldschermen. Met deze functie kunt u de instelling van het adres van de grafische kaart controleren. Als de computer een kleurenkaart heeft, wordt de logische waarde waar (.T.) weergegeven.

Voorbeeld

Om de kleuromgeving in te stellen in een programma, kunt u de volgende commando's uitvoeren in een dBASE IV-programma:

```
IF ISCOLOR()
    SET COLOR TO GR/B,W/R,GR
ELSE
    SET COLOR TO W+
ENDIF
```

Als de logische waarde waar (.T.) wordt weergegeven door de functie ISCOLOR(), wordt de weergave ingesteld voor een kleurenbeeldscherm. Als de logische waarde onwaar (.F.) wordt weergegeven, wordt de weergave ingesteld voor een monochroom beeldscherm.

Deze functie is niet betrouwbaar bij monochrome beeldschermen die kleurenbeeldschermen emuleren.

Zie ook

SET COLOR, SET DISPLAY

ISLOWER

Met de functie ISLOWER() kunt u controleren of de opgegeven tekenreeks met een kleine letter begint.

Syntaxis

ISLOWER(<Tuitdr>)

Gebruik

Met de functie ISLOWER() kunt u tekenreeksen controleren en bewerken.

Als de tekenreeks begint met een kleine letter, wordt de logische waarde waar (.T.) weergegeven. Als de tekenuitdrukking begint met een hoofdletter of een teken dat geen letter is, wordt de logische waarde onwaar (.F.) weergegeven.

Voorbeeld

U kunt als volgt controleren of een uitdrukking begint met een kleine letter:

```
. ? ISLOWER("abc123")
.T.
. ? ISLOWER("ABC123")
.F.
. ? ISLOWER("12abc")
.F.
```

Zie ook

ISALPHA(), ISUPPER(), LOWER(), UPPER()

ISMARKED()

ISMARKED() levert een logische waarde waar (.T.) op als gedurende een transactie een wijziging in een database-bestand is aangebracht.

Syntaxis

ISMARKED([<alias>])

Gebruik

Met de functie ISMARKED() kunt u in het actieve of opgegeven werkgebied kijken of de bestandsaanhef van het database-bestand een markering bevat die aangeeft dat het bestand wordt gewijzigd.

U kunt ISMARKED() na het commando BEGIN TRANSACTION gebruiken om vast te stellen of het bestand wordt gewijzigd of niet. Als de bestandsaanhef van het database-bestand aangeeft dat het bestand op dat moment wordt gewijzigd, wordt de logische waarde waar (.T.) weergegeven. Als het database-bestand niet wordt gewijzigd, wordt de logische waarde onwaar (.F.) weergegeven. Wanneer de logische waarde waar (.T.) wordt weergegeven, geeft dat aan dat een transactie nog niet is afgerond.

Als u geen alias opgeeft, wordt alleen het actieve werkgebied gecontroleerd.

Voorbeeld

Zie het voorbeeld bij de functie COMPLETED() in dit hoofdstuk.

Zie ook

BEGIN/END TRANSACTION, COMPLETED(), RESET, ROLLBACK ROLLBACK()

ISUPPER()

Met de functie ISUPPER() kunt u controleren of de opgegeven tekenuitdrukking begint met een hoofdletter.

Syntaxis

ISUPPER(<Tuitdr>)

Gebruik

Met de functie ISUPPER() kunt u tekenreeksen controleren en bewerken.

Als de opgegeven tekenreeks met een kleine letter begint, wordt de logische waarde waar (.T.) weergegeven. Als de tekenreeks begint met een kleine letter of een teken dat geen letter is, wordt de logische waarde onwaar (.F.) weergegeven.

Voorbeeld

U kunt als volgt controleren of een tekenreeks begint met een hoofdletter:

```
. ? ISUPPER("ABC123")
.T.
. ? ISUPPER("abc123")
.F.
. ? ISUPPER("12ABC")
.F.
```

Zie ook

ISALPHA(), ISLOWER(), LOWER(), UPPER()

KEY()

Met de functie KEY() kunt u de sleutelwaarde opvragen.van het indexbestand dat wordt aangeduid met <Nuitdr>.

Syntaxis

KEY([<Tuitdr>,] <Nuitdr> [,<alias>])

Gebruik

De tekenuitdrukking is een .mdx-bestand dat gerelateerd is aan het actieve database-bestand. Als u een .mdx-bestand opgeeft bij de functie, wordt de numerieke uitdrukking geïnterpreteerd aan de hand van dat .mdx-bestand. Als het opgegeven bestand niet bestaat, wordt een nultekenreeks weergegeven.

Als u geen .mdx-bestand opgeeft, worden alle geopende bestanden in het werkgebied (inclusief .ndx-bestanden) in overweging genomen voor <Nuitdr>. De .ndx-bestanden verschijnen in deze fase eerder dan de .mdx-labels.

De volgorde waarin de labels worden weergegeven:

- .ndx-bestandsuitdrukkingen (indien bestand is geopend)
- bijbehorende .mdx-labels
- niet-bijbehorende .mdx-labels

KEY() werkt in het actieve werkgebied, tenzij u het nummer van een ander werkgebied, een indirecte verwijzing naar een bestandsnaam, of de aliasnaam van een database-bestand opgeeft met de optionele parameter <alias>.

Zie ook

MDX(), NDX(), ORDER(), TAG()

LASTKEY()

Met de functie LASTKEY() kunt u de decimale ASCII-waarde opvragen van de laatst ingedrukte toets die u gebruikt hebt om een schermgroot commando uit te voeren.

Syntaxis

LASTKEY()

Gebruik

Met de functie LASTKEY() kunt u de ASCII-waarde opvragen van de toets die het laatst door de gebruiker is ingedrukt. De ASCII-waarden zijn dezelfde als die van INKEY().

Met LASTKEY() kunt u in programma's de waarde van het antwoord van de gebruiker opvragen en vervolgens reageren op basis van die waarde (bijvoorbeeld een menu of melding weergeven, of een commando of procedure uitvoeren).

Voorbeeld

In het volgende voorbeeld wordt een functietoets geprogrammeerd om een schermgroot commando te beëindigen. Daarna wordt met LASTKEY() bepaald met welke toets het commando is beëindigd:

```
SET FUNCTION 5 TO CHR(23)   && CHR(23)=Ctrl-End.
DO WHILE .T.
    * ——GET-variabelen
        .
        .
        .
    @ 19,10 SAY "Druk op F5 voor een lijst met artikelen."
    READ
    * ——Bij F5 stop.
    IF LASTKEY() = -4
        DO Artlijst
        LOOP
    ELSE
        EXIT
    ENDIF
ENDDO
```

Zie ook

INKEY(), READKEY()

LEFT()

Met de functie LEFT() kunt u een bepaald aantal tekens van een tekenuitdrukking of memoveld opvragen, te beginnen met het meest linkse teken.

Syntaxis

LEFT(<Tuitdr>/<memoveld>,<Nuitdr>)

Gebruik

Met de functie LEFT() kunt u het eerste deel van een tekenreeks of een memoveld opvragen. Deze functie levert hetzelfde resultaat op als de functie SUBSTR() met 1 als beginpositie en <Nuitdr> als het aantal tekens.

De numerieke uitdrukking bepaalt hoeveel tekens uit de tekenreeks worden gelicht. Als de numerieke uitdrukking nul is, wordt een nultekenreeks weergegeven.

Als de numerieke uitdrukking groter is dan de lengte van de tekenreeks, wordt de gehele tekenreeks weergegeven. Met LEFT() kunnen maximaal 254 tekens worden weergegeven.

Voorbeelden

Op de volgende manier worden de eerste drie tekens uit een tekenreeks gelicht:

```
. ? LEFT("abcdef",3)
abc
```

Zie ook

AT(), LTRIM(), RIGHT(), RTRIM(), STUFF(), SUBSTR(), TRIM()

LEN()

Met de functie LEN() kunt u het aantal tekens in een tekenuitdrukking of memoveld opvragen.

Syntaxis

LEN(<Tuitdr>/<memoveld>)

Gebruik

Met de functie LEN() kunt u het aantal tekens in een database-veld, geheugenvariabele of memoveld opvragen. Er wordt een numerieke waarde weergegeven die het aantal tekens weergeeft. Als het veld een nultekenreeks bevat, wordt een nul weergegeven.

Voorbeelden

Door de functie TRIM() te combineren met de functie LEN() kunt u het werkelijke aantal tekens in een database-veld berekenen:

```
. USE Afnemers
. GOTO 2
. ? Achternaam
Baljuw
. ? LEN(Achternaam)
            15
. ? LEN(TRIM(Achternaam))
            6
```

De functie LEN() werkt ook met memovelden:

```
. ? LEN(Overzicht)
            366
```

Zie ook

LTRIM(), RTRIM(), TRIM()

LIKE()

Met de functie LIKE() kunt u tekenreeksen met jokers vergelijken. Deze functie ondersteunt ook het predikaat in de set-georiënteerde SQL-taal.

Syntaxis

LIKE(<patroon>,<Tuitdr>)

Gebruik

Met de functie LIKE() kunt u de tekenreeks aan de linker kant vergelijken met de tekenreeks aan de rechter kant. De linker tekenreeks bevat jokers. De rechter tekenreeks wordt vergeleken met de linker tekenreeks. Het resultaat van deze functie is de logische waarde waar (.T.) of onwaar (.F.).

U kunt twee soorten jokers opgeven: de asterisk (*) en het vraagteken (?). De asterisk staat voor een willekeurig aantal tekens of voor geen tekens, het vraagteken staat voor één teken. Beide jokers mogen op elke plaats worden opgegeven en bovendien meer dan één keer.

De functie LIKE("*abc",eenreeks) levert de logische waarde waar (.T.) op als het veld of de geheugenvariabele *eenreeks* de tekenreeks "abc" is of op deze reeks eindigt. De functie LIKE("*abc*",eenreeks) levert de logische waarde waar op als de tekenreeks "abc" op een willekeurige plaats in *eenreeks* voorkomt.

Er wordt bij deze functie onderscheid gemaakt tussen hoofdletters en kleine letters. De jokers kunnen zowel hoofdletters als kleine letters vervangen.

Voorbeelden

Bij de functie LIKE() kunt u jokers zo vaak als u wilt en op elke plaats in de tekenreeks opgeven:

```
. ? LIKE("abr*","abracadabra")
.T.
. ? LIKE(?"a*a*","baba au rhum")
.T.
. ? LIKE("?a*a*","barbara")
.T.

. ? LIKE("?a*a*","BARBARA")
.F.
. ? LIKE("?BARBARA","BARBARA")
.F.
. ? LIKE("*BARBARA","BARBARA")
.T.
```

LINENO()

Met de functie LINENO() kunt u het regelnummer opvragen van de regel die zal worden uitgevoerd in een commando- of procedurebestand.

Syntaxis

LINENO()

Gebruik

Met de functie LINENO() kunnen programmeurs in het foutopsporingsprogramma onderbrekingspunten instellen, om een programma op een bepaalde regel te onderbreken.

Als u de functie LINENO() gebruikt in het venster Onderbrekingspunten van het foutopsporingsprogramma, wordt het venster Debugger actief en kunt u commando's intypen achter **Opdracht**.

Met de functie LINENO() kunt u in routines van ON ERROR het nummer opvragen van de regel waarop zich de fout voordeed.

Zie ook

DEBUG, PROGRAM(), SET DEVELOPMENT

LKSYS()

Met de functie LKSYS() kunt u de aanmeldingsnaam opvragen van de gebruiker die een vergrendeling heeft aangebracht bij een record of bestand dat u tevergeefs hebt geprobeerd te openen. U kunt ook de datum of de tijd opvragen waarop een record of bestand is vergrendeld. Ook de datum en het tijdstip waarop een record of bestand voor het laatst is bijgewerkt en de aanmeldingsnaam van de gebruiker die daarvoor verantwoordelijk is, kunnen worden opgevraagd.

Syntaxis

LKSYS(n)

Als n = 0 wordt de tijd van de vergrendeling weergegeven.
Als n = 1 wordt de datum van de vergrendeling weergegeven.
Als n = 2 wordt de aanmeldingsnaam weergegeven van de gebruiker die het record of bestand heeft vergrendeld.
Als n = 3 wordt het tijdstip weergegeven waarop voor het laatst is bijgewerkt of vergrendeld.
Als n = 4 wordt de datum weergegeven waarop voor het laatst is bijgewerkt of vergrendeld.
Als n = 5 wordt de aanmeldingsnaam weergegeven van de gebruiker die als laatste het record of bestand heeft bijgewerkt of vergrendeld.

Gebruik

De argumenten 0, 1 en 2 leveren alleen een waarde op nadat u hebt geprobeerd een record of bestand te vergrendelen, terwijl het record of bestand op dat moment al door een andere gebruiker is vergrendeld. Met argument 2 vraagt u de naam op van die gebruiker. De argumenten 3, 4 en 5 kunt u gebruiken ongeacht of het record of bestand al is vergrendeld of niet.

De functie levert een nultekenreeks op, tenzij met het commando CONVERT het veld _dbaselock aan het actieve database-bestand is toegevoegd. Als bij CONVERT de waarde 8 als argument wordt opgegeven, wordt de gebruikersnaam afgekapt en zendt de functie LKSYS() een nultekenreeks terug voor de argumenten 2 en 5. Geef 24 op als argument bij CONVERT als u de volledige aanmeldingsnaam wilt opvragen.

Bij het vergrendelen van een record worden tijdstip, datum en aanmeldingsnaam van de gebruiker opgenomen in het veld _dbaselock van dat record. Wanneer een database-bestand wordt vergrendeld, worden dezelfde gegevens opgenomen in het veld _dbaselock van het fysieke eerste record in het bestand. Bijwerken van de gegevens gebeurt zowel bij automatisch als expliciet vergrendelen van records en bestanden.

LKSYS()

De functie levert met de eerste drie argumenten gegevens op omtrent de laatste mislukte poging een vergrendeling aan te brengen. Bij elke mislukte poging om een geconverteerd database-bestand te vergrendelen, worden de gegevens die nodig zijn voor argumenten 0, 1 en 2 vanuit het veld _dbaselock dat op het database-bestand van toepassing is, weggeschreven in een buffer. Wanneer de functie LKSYS() met de argumenten 0, 1 of 2 wordt gebruikt, worden de gegevens uit deze buffer gelezen. Aangezien de buffer pas wordt herschreven als er opnieuw een mislukte poging tot vergrendelen wordt gedaan, levert de functie met de argumenten 0, 1 en 2 altijd de gegevens over de laatste mislukte poging tot vergrendelen op.

De laatste drie argumenten leveren gegevens op over de laatste succesvolle vergrendeling van een record of bestand. Bij de argumenten 3, 4 en 5 worden de gegevens direct uit het veld _dbaselock gehaald en niet uit een interne buffer.

Voorbeelden

Het volgende voorbeeldprogramma probeert het actieve record te vergrendelen. Wanneer het vergrendelen niet lukt, verschijnt er een melding waarin te lezen valt wie het record heeft vergrendeld en wanneer. De gebruiker kan op grond van die gegevens besluiten een nieuwe poging te doen.

```
DO WHILE .T.
    IF .NOT. RLOCK()
      ? "Record vergrendeld op" + LKSYS(1) + ;
      " om " + LKSYS(0) + "door" + LKSYS(2)
      WAIT "Wilt u het nogmaals proberen?" TO mretry
      IF UPPER(mretry) = "N"
         RETURN
      ENDIF
    ENDIF
    EXIT
ENDDO
```

In het volgende voorbeeld ziet u hoe u de datum en het tijdstip waarop een record is vergrendeld, kunt opvragen.

```
? "Record voor het laatst vergrendeld op" + LKSYS(4) + "om" +; LKSYS(3)
```

Zie ook

CHANGE(), CONVERT, FLOCK(), RLOCK(), SET LOCK, UNLOCK

LOCK()

De functie LOCK() is een atlternatieve vorm van de functie RLOCK. Raadpleeg de functie RLOCK() verderop in dit hoofdstuk.

LOG()

Met de functie LOG() kunt u het natuurlijke logaritme van een getal berekenen.

Syntaxis

LOG(<Nuitdr>)

Gebruik

Het natuurlijke logaritme heeft het grondtal e. Met de functie LOG() kunt u de exponent van de volgende vergelijking opvragen:

$y = e^x$

waarbij y de numerieke uitdrukking is die wordt opgegeven bij de functie LOG(). <Nuitdr> moet een positief geheel getal zijn. De waarde van x wordt weergegeven en de waarde is van het type F (lang, reëel getal).

Voorbeelden

Bereken het natuurlijke logaritme van e, 2.71828. Als de zoveelste macht van een getal zichzelf is, wordt een 1 weergegeven.

```
. SET DECIMALS TO 5
. ? LOG(2.71828)
                    1,00000
```

Bereken 4 * 2 met behulp van natuurlijke logaritmen en de functie EXP().

```
. SET DECIMALS TO 2
. x = 4
                    4
. y = 2
                    2
. ? LOG(x * y)
                    2,08
. ? EXP(2.08)
                    8,00
```

Zie ook

EXP(), LOG10()

LOG10()

Met de functie LOG10() kunt u het logaritme met het grondtal 10 van een opgegeven getal berekenen.

Syntaxis

LOG10(<Nuitdr>)

Gebruik

Met de functie LOG10() kunt u de waarde van y in de volgende vergelijking berekenen:

y = LOG10(x)

X is de numerieke uitdrukking die wordt opgegeven bij de functie LOG10(). De uitdrukking <Nuitdr> moet een positief geheel getal zijn. De waarde van y wordt weergegeven en is van het type F (lang, reëel getal).

Voorbeelden

```
. SET DECIMALS TO 4
. ?LOG10(2.0000)
                    0,3010
```

Bereken 2*2 met behulp van de functie LOG10():

```
. y= log10(2)+log10(2)
                    0,6021
```

Met de volgende twee uitdrukkingen kunt u de antilog berekenen:

```
. ? 10**y
                    4
```

of:

```
. ? 10^y
                    4
```

Zie ook

EXP(), LOG()

LOOKUP()

Met de functie LOOKUP() kunt u in een database-bestand een bepaald record opzoeken en de waarde van een veld in dat record opvragen.

Syntaxis

LOOKUP(<gevonden veld>,<gezochte uitdrukking>,<zoekveld>)

Gebruik

De functie zoekt in <zoekveld> naar de <gezochte uitdrukking>. Wanneer er een veld wordt gevonden dat de uitdrukking bevat, levert de functie de waarde <gevonden veld> op.

Geef een alias op als u verwijst naar uitdrukkingen buiten het actieve werkgebied.

Het database-bestand wordt sequentieel doorzocht, tenzij er een index beschikbaar is waarin <zoekveld> de sleuteluitdrukking is. U kunt de zoekactie optimaliseren door .mdx-labels en .ndx-bestanden te openen.

Aan het einde van de zoekactie staat de recordwijzer op het eerste record in het actieve database-bestand waarin <gezochte uitdrukking> overeenkomt met de waarde van <zoekveld>. Als er geen veld wordt gevonden dat voldoet aan de voorwaarde, wordt de recordwijzer aan het einde van het bestand geplaatst. De functie LOOKUP() levert de waarde op die staat in <gevonden veld>. Deze waarde kan in een uitdrukking worden gebruikt.

Voorbeeld

Het volgende is van toepassing als u informatie wilt opvragen uit een database-bestand dat niet in het actieve werkgebied staat.

Het bestand Klant bevat de namen van klanten, terwijl het bestand Delen velden met een deel-nummer voor een recente bestelling, de bestelde hoeveelheid en de over die bestelling te verlenen korting bevat.

FIND heeft betrekking op een record in het database-bestand Klant dat overeenkomt met het opgegeven klantnummer. De klant in kwestie is Jansen en heeft deel 645 besteld.

Met de functie LOOKUP() wordt in de volgende commandoregel gezocht naar het deel met het nummer 645 in het veld deel-nummer van het database-bestand Delen. Wanneer er een record overeenkomt met de gezochte uitdrukking, levert de functie de waarde op van het veld Prijs. Met de rest van de commandoregel wordt een hoeveelheid berekend met behulp van Prijs (database-bestand Delen) en Hoevlh en Korting (database-bestand Klant)

```
. SELECT 1
. USE Delen
. SELECT 2
. USE Klant ORDER TAG Klntnr
. FIND 1130
. ? Achternm, Deelnr
Jansen              645
. pline 2 = LOOKUP(Delen->Prijs, Deelnr, Delen->Deelnr) * Hoevlh - (Hoevlh * Korting)
```

LOWER()

Met de functie LOWER() kunt u hoofdletters omzetten in kleine letters. De functie kan niet worden gebruikt voor een memoveld, tenzij eerst met de functie MLINE() een tekenreeks uit het memoveld is verwijderd.

Syntaxis

LOWER(<Tuitdr>)

Voorbeelden

```
. ? LOWER("DIT IS EEN MOOIE DAG")
dit is een mooie dag

. STORE "DIT IS EEN MOOIE DAG" TO mooiedag
DIT IS EEN MOOIE DAG
. ? mooiedag
DIT IS EEN MOOIE DAG

? mooiedag = "dit is een mooie dag"
.F.

. ? LOWER(mooiedag)= "dit is een mooie dag"
.T.
```

Als u het eerste teken van een tekenreeks niet in een kleine letter wilt omzetten:

```
. tekenreeks = "AMSTERDAM"
AMSTERDAM
. nieuwreeks = LEFT(tekenreeks,1) + SUBSTR(LOWER(tekenreeks),2)
Amsterdam
```

Zie ook

ISALPHA(), ISLOWER(), ISUPPER(), MLINE(), UPPER()

LTRIM()

Met de functie LTRIM() kunt u de voorloopspaties van een tekenreeks verwijderen.

Syntaxis

LTRIM(<Tuitdr>)

Gebruik

Met de functie LTRIM() kunt u voorloopspaties verwijderen.

Voorbeeld

In het volgende voorbeeld worden de voorloopspaties verwijderd van een getal dat is omgezet in een tekenreeks:

```
. ? STR(11.95,8,2)
      11,95
. ? LTRIM(STR(11.95,8,2))
11,95
```

Zie ook

LEFT(), RIGHT(), RTRIM(), STR(), SUBSTR()

LUPDATE()

Met de functie LUPDATE() kunt u opvragen op welke datum het opgegeven bestand voor het laatst is bijgewerkt.

Syntaxis

LUPDATE([<alias>])

Gebruik

Met de functie LUPDATE() kunt u opvragen wanneer het actieve database-bestand voor het laatst is bijgewerkt. Als er geen database-bestand actief is, wordt een blanco datum weergegeven.

Als u geen alias opgeeft, wordt de datum van het bestand in het actieve werkgebied weergegeven. Als alias kunt u opgeven: het nummer van een werkgebied, de aliasnaam van een werkgebied, of een indirecte verwijzing naar een bestand die door het programma kan worden geïnterpreteerd als de naam van een database-bestand.

Voorbeelden

Met de volgende regels kunt u er in een programma voor zorgen dat recu's slechts één keer per dag worden ingevoerd:

```
IF LUPDATE() < DATE()
    Do_invoer = "?"
    @ 5,1 SAY "Recu's zijn voor het laatst ingevoerd op " + DTOC(LUPDATE())+;
    "Invoeren? (J/N)" GET Do_invoer PICTURE "J"
    READ
    IF Do_invoer = "J"
        DO <invoerprocedure>
    ENDIF
ENDIF
```

Zie ook

DBF(), DTOC(), FIELD(), MDX(), NDX(), ORDER(), RECCOUNT(), RECSIZE(), TAG()

MAX()

De functie MAX() levert de grootste van twee numerieke uitdrukkingen, of datum- of tekenuitdrukkingen op.

Syntaxis

MAX(<uitdr1>,<uitdr2>)

Gebruik

Met de functie MAX() kunt u twee numerieke uitdrukkingen, datumuitdrukkingen of tekenuitdrukkingen vergelijken om te bepalen welke van de twee de grootste is. Als datumuitdrukkingen worden vergeleken, wordt de laatste van de twee datums teruggezonden.

Deze functie verschilt van de verzamelfunctie MAX van het commando CALCULATE. Raadpleeg het gedeelte over het commando CALCULATE voor een bespreking van de verzamelfunctie MAX.

Voorbeeld

U kunt als volgt de grootste van twee waarden bepalen:

```
. eerste = 12.32
            12,32
. tweede = 34.12
            34,12
. ? MAX(eerste, tweede)
            34,12
```

Op de volgende manier kunt u bepalen welke van twee tekenuitdrukkingen de grootste is:

```
. Naam = "Manes"
Manes
. ? MAX(Naam,"Smits")
Smits
```

Zie ook

CALCULATE, MIN()

MDX()

Met de functie MDX() kunt u de bestandsnaam opvragen van het .mdx-bestand dat op de opgegeven positie in de indexlijst staat.

Syntaxis

MDX[<Nuitdr >[,<alias>])

Gebruik

Met de functie MDX() kunt u de bestandsnaam opvragen van het actieve .mdx-bestand in de .mdx-bestandenlijst van het actieve werkgebied. De numerieke uitdrukking <Nuitdr> geeft de positie weer van het .mdx-bestand in de bestandenlijst die is opgegeven met het commando SET INDEX TO <lijst indexbestanden>. Als er geen bestandenlijst is opgegeven of geen .mdx-bestand is gekoppeld aan de opgegeven waarde, wordt een nultekenreeks teruggezonden.

Als u geen alias opgeeft, wordt uitgegaan van het actieve werkgebied.

Zie ook

KEY(), NDX(), ORDER(), SET FULLPATH, TAG()

MDY()

Met de functie MDY() kunt u de indeling van datums wijzigen in maand dag, jaar.

Syntaxis

MDY(<Duitdr>)

Gebruik

Met de functie MDY() kunt u de indeling van elke datum wijzigen in maand (volledige naam van de maand) dag (twee cijfers), jaar (twee cijfers). Als SET CENTURY op ON is ingesteld, wordt het jaar weergegeven met vier cijfers.

De datum wordt teruggezonden in de vorm van een tekenuitdrukking.

Voorbeeld

U kunt de datum 29-02-88 als volgt weergeven met de functie MDY():

```
. schrikdat = {29-02-88}
29-02-88
. ? schrikdat
29-02-88
. SET CENTURY ON
. ? MDY(schrikdat)
29 februari 1988
```

Zie ook

DMY(), SET CENTURY, SET DATE

MEMLINES()

Met de functie MEMLINES() kunt u het aantal regels in een memoveld opvragen, aangenomen dat het memoveld is uitgevuld volgens de instelling SET MEMOWIDTH.

Syntaxis

MEMLINES(<naam memoveld>)

Gebruik

Met de functie MEMLINES() kunt u het aantal regels in een memoveld berekenen aan de hand van de met SET MEMOWIDTH ingestelde breedte van het memoveld.

Voorbeeld

In dit voorbeeld wordt het verband geïllustreerd tussen het commando SET MEMOWIDTH TO en het aantal regels dat de functie MEMLINES() oplevert. Met de instelling van SET MEMOWIDTH wordt bepaald op welke positie een regel wordt afgebroken en dus ook hoeveel regels tekst er in het memo staan.

Stel dat in het memoveld Jantje_mem de volgende tekst is opgeslagen:

Jantje zag eens pruimen hangen, oh, als eieren zo groot.

```
. SET MEMOWIDTH TO 20
. ? Jantje_mem
Jantje zag eens
pruimen hangen, oh,
als eieren zo groot.
. ? MEMLINES(Jantje_mem)
3
. SET MEMOWIDTH TO 35
. ? Jantje_mem
Jantje zag eens pruimen hangen, oh,
als eieren zo groot.
. ? MEMLINES(Jantje_mem)
2
```

Zie ook

MLINE(), SET MEMOWIDTH

MEMORY()

Met de functie MEMORY() kunt u opvragen hoeveel RAM-geheugen er nog beschikbaar is.

Syntaxis

MEMORY()
of
MEMORY(0)

Gebruik

Met de functie MEMORY() kunt u opvragen hoeveel RAM-geheugen er nog beschikbaar is, uitgedrukt in kilobytes. Eén kilobyte is 1.024 bytes.

Met de functie MEMORY() kunt u controleren of er nog genoeg geheugen beschikbaar is, voordat u toepassingen uitvoert die extra geheugen nodig hebben voor menu's, vensters en reeksen. Met deze functie kunt u ook controleren of er nog genoeg geheugen beschikbaar is om programma's of DOS-commando's uit te voeren met het commando RUN/!.

U hoeft de parameter 0 niet op te geven bij de functie MEMORY(), aangezien altijd dezelfde waarde wordt teruggezonden.

Voorbeeld

In dit voorbeeld wordt met de functie MEMORY() gecontroleerd of er nog voldoende geheugen beschikbaar is om Command.com te laden, voordat het commando RUN wordt gegeven:

```
IF MEMORY() < 25
    ? Er is niet genoeg geheugen beschikbaar voor Command.com."
ELSE
    RUN COPY Rekn_rec.dbf A:
ENDIF
```

Zie ook

DISKSPACE(), GETENV(), OS()

MENU()

Met de functie MENU() kunt u de naam van het actieve menu opvragen.

Syntaxis

MENU()

Gebruik

Met de functie MENU() kunt u de naam opvragen van het menu dat het laatst is geactiveerd. Als er geen menu actief is, wordt een nultekenreeks teruggezonden.

Met deze functie kunt u de naam opvragen van het menu dat u hebt gebruikt voordat u de hulpfunctie opriep met de **F1**-toets.

Voorbeeld

Als het menu Diversen als laatste is geactiveerd en nog steeds actief is, wordt dat bevestigd door de functie MENU().

```
. ? MENU()
Diversen
```

Zie ook

ACTIVATE MENU, DEFINE MENU

MESSAGE()

Met de functie MESSAGE() kunt u de foutmelding opvragen van de fout die een ON ERROR-conditie heeft veroorzaakt.

Syntaxis

MESSAGE()

Gebruik

De foutmelding die wordt teruggezonden kan worden weergegeven op het scherm of worden opgeslagen in een variabele.

Raadpleeg bijlage A voor een lijst met alle foutmeldingen die kunnen worden weergegeven als dBASE individueel wordt gebruikt.

De foutmeldingen van dBASE IV die specifiek zijn voor een netwerk, zijn te vinden in het boek *dBASE IV in netwerken*.

Zie ook

CERROR(), ERROR(), ON ERROR

MIN()

Met de functie MIN() kunt u twee numerieke uitdrukkingen, datumuitdrukkingen of tekenuitdrukkingen met elkaar vergelijken. De kleinste van de twee uitdrukkingen wordt teruggezonden.

Syntaxis

MIN(<uitdr1>,<uitdr2>)

Gebruik

Met de functie MIN() kunt u de kleinste van twee uitdrukkingen opvragen. Wanneer twee datumuitdrukkingen worden vergeleken, wordt de vroegste datum teruggezonden.

Deze functie verschilt van de verzamelfunctie MIN() van het commando CALCULATE. Raadpleeg het gedeelte over het commando CALCULATE voor een bespreking van de verzamelfunctie MIN().

Voorbeelden

U kunt als volgt bepalen welke van twee uitdrukkingen het kleinst is:

```
. eerste = 123.54
          123,54
. tweede = 232.63
          232,63
. ? MIN(eerste, tweede)
          123,54
. ? IIF(MIN(eerste, tweede) = eerste, "Eerste" , "Tweede") + " is kleiner."
Eerste is kleiner.
```

U kunt ook de kleinste van twee tekenuitdrukkingen bepalen:

```
. Naam = "Manes"
Manes
. ? MIN(Naam, "Smits")
Manes
```

Zie ook

CALCULATE, MAX()

MLINE()

Met de functie MLINE() kunt u de opgegeven regel uit een memoveld in het actieve record lichten.

Syntaxis

MLINE(<memoveld>,<Nuitdr>)

Gebruik

De opgegeven tekstregel wordt uit het memoveld gelicht. Aan de hand van de marges die zijn ingesteld met SET MEMOWIDTH wordt bepaald welke tekst op de opgegeven regel staat.

Voorbeeld

In het volgende voorbeeld wordt het verband geïllustreerd tussen het commando SET MEMOWIDTH TO en de functie MLINE(). Met SET MEMOWIDTH TO kunt u bepalen op welke positie een regel wordt afgebroken en dus ook welke regel wordt weergeven door de functie MLINE().

Stel dat in het memoveld Jantje_mem de volgende tekst is opgeslagen:

Jantje zag eens pruimen hangen, oh, als eieren zo groot.

```
. SET MEMOWIDTH TO 15
. ? Jantje_mem
Jantje zag eens
pruimen hangen,
oh, als eieren
zo groot.
. ? MLINE(Jantje_mem,2)
pruimen hangen,
. SET MEMOWIDTH TO 35
. ? Jantje_mem
Jantje zag eens pruimen hangen,
oh, als eieren zo groot.
. ?MLINE(Jantje_mem,2)
oh, als eieren zo groot.
```

Zie ook

MEMLINES(), SET MEMOWIDTH

MOD()

Met de functie MOD() kunt u de rest opvragen van een deling van twee numerieke uitdrukkingen. MOD() is vooral handig voor het omzetten van eenheden, bijvoorbeeld centimeters in inches.

Syntaxis

MOD(<Nuitdr1>,<Nuitdr2>)

Gebruik

De functie MOD() zendt een geheel getal terug, de *modulus*. Dit getal is de rest van de deling van de uitdrukking <Nuitdr1> door de uitdrukking <Nuitdr2>.

Als <Nuitdr2> positief is, wordt een positief getal teruggezonden. Als <Nuitdr2> negatief is, wordt een negatief getal teruggezonden.

De formule voor de modulus is:

<Nuitdr1> - FLOOR(<Nuitdr1>/<Nuitdr2>) * <Nuitdr2>

FLOOR is een rekenkundige functie die het grootste gehele getal terugzendt dat kleiner dan of gelijk is aan het argument.

Voorbeelden

U kunt op de volgende manier de modulus van twee getallen berekenen:

```
. ? MOD(14,12)
                2

. ? MOD(0,32)
                0

. ? MOD(1,-3)
               -2
```

Zie ook

FLOOR(), INT()

MONTH()

Met de functie MONTH() kunt u de maand opvragen van een datumuitdrukking. De maand wordt teruggezonden in de vorm van een getal.

Syntaxis

MONTH(<Duitdr>)

Gebruik

De datumuitdrukking kan een geheugenvariabele of een veld zijn, of een functie die een datum oplevert.

Voorbeelden

Als 15-05-87 de systeemdatum is:

```
. ? MONTH(DATE())
                5
```

In het volgende voorbeeld wordt de maand van de systeemdatum opgeslagen in de geheugenvariabele Mmaand:

```
. STORE MONTH(DATE()) TO Mmaand
                5

. ? Mmaand
                5
```

Zie ook

CMONTH(), DAY(), YEAR()

NDX()

Met de functie NDX() kunt u de naam van een .ndx-bestand opvragen.

Syntaxis

NDX(<Nuitdr> [,<alias>])

<Nuitdr> is de volgorde van de index, of een willekeurig getal als ORDER niet is opgegeven. De alias kan het cijfer of de letter van een werkgebied zijn, de aliasnaam van het geopende database-bestand of een indirecte verwijzing naar een bestandsnaam.

Gebruik

Met de functie NDX() kunt u in een programma geopende indexbestanden wijzigen zonder de namen van de bestanden te kennen.

De numerieke uitdrukking is de positie van het geopende indexbestand in de indexbestandenlijst van het actieve werkgebied. Als de gebruiker een indexbestandenlijst heeft opgegeven met SET INDEX TO <lijst indexbestanden> of USE INDEX <lijst indexbestanden>, kan de numerieke uitdrukking de waarde 1 tot en met n hebben, waarbij n het aantal indexbestanden in de indexbestandenlijst is.

De functie NDX() werkt met het database-bestand in het actieve werkgebied. De naam van het indexbestand wordt teruggezonden, afhankelijk van het getal dat de gebruiker opgeeft bij <Nuitdr>.

Als <Nuitdr> niet verwijst naar een index, wordt een nultekenreeks teruggezonden.

Als de aliasnaam niet bestaat, wordt de foutmelding ALIAS niet gevonden gegeven.

Als de optionele aliasnaam niet wordt opgegeven, wordt uitgegaan van het actieve werkgebied.

Zie ook

ALIAS(), DBF(), FIELD(), MDX(), ORDER(), RECCOUNT(), RECSIZE(), SET FULLPATH, SET INDEX, SET ORDER

NETWORK

Met de functie NETWORK() kunt u bepalen of het systeem in een netwerk draait. De logische waarde waar (.T.) of onwaar (.F.) is het resultaat.

Syntaxis

NETWORK()

Gebruik

Met de functie NETWORK() kunt u in programma's bepalen of dBASE IV in een netwerk wordt gebruikt en aan de hand daarvan bepaalde sprongopdrachten laten uitvoeren.

Voorbeeld

```
. ? NETWORK()
.F.
```

Zie ook

GETENV(), OS()

ORDER()

Met de functie ORDER() kunt u de naam opvragen van de .mdx-label of het indexbestand dat de hoofdvolgorde bepaalt.

Syntaxis

ORDER([<alias>])

De optie <alias> is de aliasnaam van een geopend database-bestand.

Gebruik

Met de functie ORDER() kunt u de naam opvragen van de index die de indexvolgorde bepaalt van het actieve database-bestand, of het database-bestand dat wordt opgegeven met de alias.

Het hoofddeel van de bestandsnaam van het .ndx-bestand wordt in hoofdletters teruggezonden. Als er geen .ndx-bestand is, wordt de naam van de .mdx-label teruggezonden, eveneens in hoofdletters.

Als u geen alias opgeeft, wordt uitgegaan van het actieve werkgebied.

Voorbeeld

```
. USE Afnemers ORDER Klantnaam
Hoofdindex: KLANTNAAM
. ? ORDER("Klantnaam")
KLANTNAAM
. SET ORDER TO TAG Klant_nr
Hoofdindex: KLANT_NR
. ? ORDER()
KLANT_NR
```

Zie ook

KEY(), MDX(), NDX(), SET INDEX, SET ORDER, TAG(), USE

OS()

Met de functie OS() kunt u de naam van het besturingssysteem opvragen waaronder dBASE IV draait.

Syntaxis

OS()

Gebruik

Met de functie OS() kunt u de naam van het besturingssysteem opvragen waaronder dBASE IV draait.

Voorbeeld

```
<EXAMPLE>.<CUR>?OS()<SIEF>
DOS 3.10
```

Zie ook

GETENV(), NETWORK(), VERSION()

PAD()

Met de functie PAD() kunt u de naam opvragen van de strip die als laatste is gekozen in het actieve menu.

Syntaxis

PAD()

Gebruik

Wanneer een menu actief is, kan de cursor op één van de opties worden geplaatst. Elke optie wordt weergegeven op een strip. Als u een optie kiest door op ↵ te drukken, wordt deze strip de laatst gekozen strip. Met de functie PAD() kunt u de naam van deze strip opvragen.

De keuze die met het commando PAD() is opgeslagen, wordt niet gewist door DEACTIVATE MENU of DEACTIVATE POPUP.

Voorbeeld

Stel dat er een menu is met de naam Diversen en dat dit menu drie strips bevat, Weergeven, Sorteren en Opslaan. Als het menu actief is, kunt u met de functie PAD() opvragen welke keuzestrip als laatste is gekozen:

```
. ? PAD()
Sorteren
```

Zie ook

DEFINE BAR, DEFINE MENU, DEFINE PAD, DEFINE POPUP, MENU(), ON PAD, ON SELECTION PAD, POPUP()

Met de functie PAYMENT() kunt u de vaste, regelmatige betaling berekenen die nodig is om

PAYMENT()

een lening met een vaste rentevoet in een bepaald aantal termijnen af te lossen.

Syntaxis

PAYMENT(<hoofdsom>,<rente>,<termijn>)

<hoofdsom> is een numerieke uitdrukking die het beginbedrag van de lening weergeeft. De uitdrukking kan zowel negatief als positief zijn.

<rente> is een positief decimaal getal dat de rentevoet weergeeft. Wanneer u een maandelijkse betaling berekent en daarbij een rentevoet op jaarbasis gebruikt, moet u het bedrag aan rente door twaalf delen.

<termijn> is een positieve numerieke uitdrukking die het aantal periodieke betalingen weergeeft. Breuken worden afgerond op een geheel getal.

Gebruik

Met de functie PAYMENT() kunt u het bedrag berekenen dat per periode moet worden betaald om de hoofdsom en de rente over een bepaald aantal termijnen af te lossen.

Voorbeeld

In het volgende voorbeeld wordt de maandelijkse betaling berekend voor een lening van 7.500,- met een looptijd van 3 jaar tegen een rente van 1,5%:

```
. INPUT "Voer de hoofdsom in: " TO hoofdsom
Voer de hoofdsom in: 7500,00
. INPUT "Voer de rente in: " TO rente
Voer de rente in: 0,015
. INPUT "Voer het aantal termijnen in: " TO termijn
Voer het aantal termijnen in: 36
. ? PAYMENT(hoofdsom, rente, termijn)
271,14
```

Zie ook

CALCULATE, FV(), PV()

PCOL()

Met de functie PCOL() kunt u opvragen in welke kolom de printkop staat (relatief aan _ploffset). Met deze functie kunt u in programma's bijhouden op welke positie de printkop zich bevindt.

Syntaxis

PCOL()

Gebruik

Met de functie PCOL() kunt u de printer relatief adresseren. Bij de commando's @...SAY en @...GET kan met de speciale operator $ worden opgegeven waar op de pagina de tekst moet worden afgedrukt. Bijvoorbeeld:

@ 1,PCOL() + 5 is hetzelfde als
@ 1, $ + 5.

Met beide opdrachten wordt de printkop vijf kolommen naar rechts verplaatst.

De waarde die wordt teruggezonden door PCOL() is relatief aan de instelling van _ploffset. Dat betekent dat de waarde van _ploffset wordt opgeteld bij de waarde van de printermarge.

De functie PCOL() werkt alleen voor de printer. Met de speciale operator $ kan zowel de uitvoer naar het scherm als de uitvoer naar de printer worden bestuurd.

Voorbeeld

In het volgende programma wordt bepaald op welke rij de printkop staat:

```
SET DEVICE TO PRINTER
@ PROW(), PCOL(), SAY "TEST"
? PCOL()     4
SET DEVICE TO SCREEN
RETURN
```

Zie ook

@, COL(), PROW(), ROW(), SET MARGIN, _pcolno, _ploffset

PI()

Met de functie PI() kunt u het irrationele getal 3,14159 opvragen. Dit getal is een benadering van de constante die de verhouding weergeeft tussen de omtrek en doorsnede van een cirkel.

Syntaxis

PI()

Gebruik

De constante Pi wordt gebruikt in wiskundige en bouwtechnische berekeningen. Met de commando's SET DECIMALS en SET PRECISION kunt u bepalen hoe nauwkeurig het getal wordt weergeven.

U kunt meer cijfers achter de komma weergeven door de waarde van SET DECIMALS te verhogen.

Voorbeelden

```
. ? PI()
                3,14
```

Bereken het oppervlak van een cirkel met de doorsnede 1. Het oppervlak is altijd Pi en de uitkomst wordt altijd uitgedrukt in de meetkundige eenheid waarmee de diameter is gemeten, bijvoorbeeld kilometers, inches of mijlen.

```
. r = 1
                1
. a = PI() * r * r
                3,14
```

POPUP()

Met de functie POPUP() kunt u de naam van het actieve pop-up menu opvragen.

Syntaxis

POPUP()

Gebruik

Het actieve pop-up menu is het pop-up menu dat als laatste is geactiveerd en nog niet inactief is gemaakt. Als er geen pop-up menu actief is, wordt een nultekenreeks teruggezonden.

Ook wanneer er geen pop-up menu is gedefinieerd, levert de functie een nultekenreeks op.

Voorbeeld

Stel dat het pop-up menu POP1 actief is:

```
. ? POPUP()
POP1
```

Zie ook

ACTIVATE POPUP, BAR(), DEFINE POPUP, MENU(), PAD(), PROMPT()

PRINTSTATUS

Met de functie PRINTSTATUS() kunt u controleren of de printer in staat is gegevens te ontvangen. Als de printer gereed is, resulteert de functie in de logische waarde waar (.T.).

Syntaxis

PRINTSTATUS()

Gebruik

Met PRINTSTATUS() kunt u de status van de laatst gekozen printer controleren, ongeacht de wijze waarop die printer is gekozen. U kunt een printer op verschillende manieren kiezen: met het commando SET PRINTER ON, met het commando SET DEVICE TO PRINTER, met Ctrl-P of met elk ander commando waarbij de clausule TO PRINTER kan worden opgegeven.

WAARSCHUWING
De functie levert alleen een waarde op als de printer rechtstreeks op uw computer is aangesloten. Bij printers die via een netwerk worden aangestuurd of via een ander apparaat aan uw computer zijn aangesloten, is het mogelijk dat de functie geen juiste respons terugstuurt.

Voorbeeld

In het volgende voorbeeld wordt de status van de printer gecontroleerd voordat een rapport wordt afgedrukt:

```
Mgereed = ""
DO WHILE .T.
    @ 23,10 SAY "Maak de printer gereed voor het rapport."
    @ 24,10 SAY "Druk op P om het rapport af te drukken,"
    @ 25,10 SAY "elke andere toets maakt de opdracht ongedaan".
                GET Mgereed PICTURE "!"
    READ
    IF .NOT. PRINTSTATUS() .AND. Mgereed = "P"
                LOOP
    ENDIF
    IF Mgereed = "P"
                REPORT FORM Personel TO PRINT
    ENDIF
    EXIT
ENDDO
```

Zie ook

SET DEVICE, SET PRINTER

PROGRAM()

Met de functie PROGRAM() kunt u de naam opvragen van het programma, de procedure of de door de gebruiker gedefinieerde functie die werd uitgevoerd toen zich een fout voordeed.

Syntaxis

PROGRAM()

Gebruik

De naam van het programmabestand of de procedure wordt teruggezonden in de vorm van een tekenreeks.

U kunt deze functie gebruiken in de vensters Onderbrekingspunt en Weergave van Debug, of in een programma. Als een programma is onderbroken, kunt u deze functie ook oproepen vanaf de commandostip. Daarnaast kan PROGRAM() worden gebruikt in combinatie met ON ERROR om de naam van het programma vast te stellen dat werd uitgevoerd toen de fout zich voordeed. Als u de functie oproept vanaf de commandostip en er geen programma is onderbroken, wordt een nultekenreeks teruggezonden in plaats van een programmanaam.

De programmanaam die wordt teruggezonden, bevat geen extensie.

Als de fout zich heeft voorgedaan in een procedure, wordt de naam van de procedure teruggezonden, niet de naam van het procedurebestand.

U hoeft geen parameters op te geven bij de functie PROGRAM().

Voorbeeld

Als de volgende regel wordt opgenomen in een programma, worden de naam en het regelnummer teruggezonden van het programma- of procedurebestand waarin de fout zich heeft voorgedaan:

```
ON ERROR ? "De fout zit in " + PROGRAM() + "op regel" + LTRIM(STR(LINENO()))
```

Zie ook

DEBUG, LINENO(), ON ERROR, RESUME, SET DEVELOPMENT, SUSPEND

PROMPT()

Met de functie PROMPT() kunt u de aanwijzing opvragen van de laatst gekozen optie van een menu of pop-up.

Syntaxis

PROMPT()

Gebruik

Met de functie PROMPT() kunt u de tekenreeks opvragen die is verbonden aan de laatst gekozen optie van een menu of pop-up.

Als een menu inactief is gemaakt doordat de gebruiker op **Esc** heeft gedrukt, wordt een nultekenreeks teruggezonden.

Als er geen pop-up menu ander menu actief is, wordt een nultekenreeks teruggezonden. Een pop-up menu of ander menu dat inactief is gemaakt, levert bij deze functie nog steeds een waarde op.

Als de aanwijzing is gedefinieerd met het commando DEFINE POPUP (daar vallen ook de PROMPT-opties van FIELD, FILES en STRUCTURE onder), wordt een andere tekenreeks teruggezonden voor elke PROMPT-optie:

FIELD: — De inhoud van het veld van de database wordt teruggezonden
FILES — De complete bestandsnaam met pad en extensie wordt teruggezonden in hoofdletters
STRUCTURE — De veldnaam wordt teruggezonden in hoofdletters

Voorbeeld

Met de functie PROMPT() kunt u in een programmabestand de aanwijzing laten weergeven op de informatieregel, zodat de gebruiker kan zien welke keuze is gemaakt:

```
DEFINE POPUP Hoofd FROM 5,10 TO 12,30
DEFINE BAR 1 OF Hoofd PROMPT "Records bewerken"
DEFINE BAR 3 OF Hoofd PROMPT "Records toevoegen"
DEFINE BAR 5 OF Hoofd PROMPT "Stop"
.
.
.
ON SELECTION POPUP Hoofd SET MESSAGE TO PROMPT()
SET STATUS ON
ACTIVATE POPUP Hoofd
```

Zie ook

BAR(), DEFINE POPUP, ON SELECTION PAD, ON SELECTION POPUP, POPUP()

PROW()

Met de functie PROW() kunt u opvragen op welke rij de printkop staat.

Syntaxis

PROW()

Gebruik

Met de functie PROW() kunt u de printer relatief adresseren. Wanneer een blok tekst niet over twee pagina's mag worden verdeeld, kunt u met de functie PROW() controleren of er nog genoeg ruimte op de pagina is. Als dat niet het geval is, kunt u eerst een pagina-onderbreking naar de printer zenden.

De waarde van PROW() is 0 als de printkop aan het begin van een nieuwe pagina staat.

Met PROW() worden geen rij- en kolomposities binnen een afdrukbestand geadresseerd.

Voorbeeld

In het volgende voorbeeld wordt de zin *Een voorbeeld van relatief adresseren* vijf regels onder de huidige positie van de printkop afgedrukt:

```
. SET DEVICE TO PRINTER
. @ PROW()+5,1 SAY "Een voorbeeld van relatief adresseren"
. SET DEVICE TO SCREEN
```

Zie ook

@, COL(), EJECT, PCOL(), ROW(), SET DEVICE, SET PRINTER, _plineno

PV()

Met de functie PV() kunt u de huidige waarde berekenen van vaste, regelmatige investeringen tegen een vaste rente over een bepaald aantal termijnen.

Syntaxis

PV(<bedrag>,<rente>,<termijn>)

Gebruik

Met de functie PV() kunt u de huidige waarde berekenen van geïnvesteerde gelden. Met deze functie kan worden berekend hoeveel moet worden geïnvesteerd om in de toekomst een bepaalde waarde te hebben geaccumuleerd.

<bedrag> is een numerieke uitdrukking die een constante regelmatige betaling weergeeft. De uitdrukking kan zowel positief als negatief zijn.

<rente> is een positieve decimale uitdrukking die de rentevoet weergeeft. Als de rente per jaar wordt berekend, moet de jaarrente worden gebruikt als een decimaal getal. Als de rente over kortere perioden wordt berekend, moet de rente per jaar worden omgerekend:

Termijnen van een dag = rente per jaar/365
Termijnen van een maand = rente per maand/12

<termijn> is het aantal betalingen, afgerond op een geheel getal.

Voorbeeld

```
. ? PV(1,.06,5)
        4,21
```

Zie ook

FV(), PAYMENT()

RAND()

Met de functie RAND() kunt u een willekeurig getal genereren.

Syntaxis

RAND([<Nuitdr>])

Gebruik

Met de functie RAND() kunt u een willekeurig getal genereren, al dan niet aan de hand van een numeriek argument. Om volgende willekeurige getallen uit de reeks te laten genereren moet u de functie herhalen zonder parameters op te geven.

De numerieke uitdrukking wordt gebruikt om een nieuw willekeurig getal te genereren. Als de uitdrukking een negatief getal is, wordt met behulp van de systeemklok een willekeurig getal gegenereerd. De uitdrukking wordt herleid tot een geheel getal zodat RAND(1,1) en RAND(1) dus dezelfde waarde opleveren. RAND() en RAND(0) zijn gelijkwaardig.

De uitkomst van deze functie heeft een bereik van 0 tot en met 0,999999.

Standaard wordt het getal 100001 gebruikt om een willekeurig getal te berekenen. Met RAND(100001) kunt u de reeks willekeurige getallen opnieuw op de standaardwaarde instellen.

Voorbeeld

```
. ? RAND(23)
                0,13
. ? RAND()      &&het volgende willekeurige getal in de reeks
```

READKEY()

Met de functie READKEY() kunt u opvragen met welke toets een schermgroot commando is beëindigd. Er wordt een geheel getal teruggezonden dat de toets aangeeft. Als er tijdens het schermgrote commando gegevens zijn gewijzigd, wordt een ander getal teruggezonden.

Syntaxis

READKEY()

Gebruik

Met de functie READKEY() kan in een programma worden bepaald welke handelingen moeten worden uitgevoerd nadat een schermgroot commando is beëindigd. De schermgrote commando's zijn APPEND, BROWSE, CHANGE, CREATE, EDIT, INSERT, MODIFY en READ.

Tabel 4-3 bevat de waarden die door READKEY() worden teruggezonden als een schermgroot commando op een bepaalde manier is beëindigd. Voor iedere manier van beëindigen zijn twee codes mogelijk. Als de gegevens niet zijn gewijzigd, wordt een code tussen 0 en 36 teruggezonden. Als de gegevens wel zijn gewijzigd, wordt de code van de toets verhoogd met 256.

Tabel 4-3 codes van READKEY()

Niet bijgewerkt Code	Bijgewerkt Code	Ingedrukte toets	Functie van toets
0	256	**Ctrl**-S, ←, **Ctrl**-H	één teken achteruit
—	256	**Spatie terug**	één teken achteruit
1	257	**Ctrl**-D, →, **Ctrl**-L	één teken vooruit
2	258	**Ctrl**-A, **Ctrl**-←	aan begin voorgaande woord
3	259	**Ctrl**-F, **Ctrl**-→	aan begin volgende woord
4	260	**Ctrl**-E, **Ctrl**-K, ↑	één veld achteruit
5	261	**Ctrl**-J, **Ctrl**-X, ↓	één veld vooruit
6	262	**Ctrl**-R, **PgUp**	één scherm achteruit
7	263	**Ctrl**-C, **PgDn**	één scherm vooruit
12	—	**Ctrl**-Q, **Esc**	beëindigen zonder opslaan
—	270	**Ctrl**-W, **Ctrl-End**	beëindigen en opslaan

(wordt vervolgd)

Tabel 4-3 codes van READKEY() (vervolg)

Niet bijgewerkt Code	Bijgewerkt Code	Ingedrukte toets	Functie van toets
15	271	↵, Ctrl-M	RETURN of vul laatste record
16 APPEND	—	↵, Ctrl-M	aan het begin van een record in
33	289	Ctrl-Home	menu aan- en uitzetten
34	290	Ctrl-PgUp	uitzoomen
35	291	Ctrl-PgDn	inzoomen
36	292	F1	Hulpfunctietoets

Voorbeeld

In het volgende voorbeeld wordt bepaald of een geheugenvariabele is gewijzigd en wordt een veld vervangen als dat het geval is:

```
IF READKEY() >= 256      && Gegevens zijn gewijzigd
REPLACE Naam WITH Mnaam
ENDIF
```

Zie ook

INKEY(), LASTKEY(), ON KEY, READ

RECCOUNT()

Met de functie RECCOUNT() kunt u opvragen hoeveel records er in het opgegeven database-bestand staan.

Syntaxis

RECCOUNT([<alias>])

Gebruik

De functie RECCOUNT is een snel alternatief voor het commando GO BOTTOM, gevolgd door de functie RECNO(). Met deze functie kunt u snel alle records in een bestand tellen.

Als er geen database-bestand actief is, wordt een nul teruggezonden.

Alle records worden geteld door de functie RECCOUNT(), zelfs als SET DELETED op ON is ingesteld en voorwaarden zijn ingesteld met het commando SET FILTER.

Richtlijnen voor het programmeren

Combineer de functie RECCOUNT() met de functies RECSIZE() en DISKSPACE() in toepassingsprogramma's waarin automatisch een reservekopie wordt gemaakt van een database-bestand. Met deze functies kunt u controleren of er nog voldoende ruimte op het station is voor de reservekopie. Zie het voorbeeld bij de functie RECSIZE() in dit hoofdstuk.

Voorbeeld

U kunt als volgt de records in het database-bestand Afnemers tellen:

```
. USE Afnemers
. ? RECCOUNT()
    8
```

Zie ook

DBF(), DISKSPACE(), RECSIZE()

RECNO()

Met de functie RECNO() kunt u het actieve record van het opgegeven bestand opvragen.

Syntaxis

RECNO([<alias>])

Gebruik

De volgende waarden worden teruggezonden door de functie RECNO():

Geen records in een database-bestand: RECNO() = 1
Geen database-bestand in gebruik: RECNO() = 0

Als de recordwijzer voorbij het laatste record in het bestand wordt geplaatst (EOF() is waar), is de teruggezonden waarde het aantal records in het bestand plus één.

Als de recordwijzer voor het eerste record in het bestand wordt geplaatst (BOF() is waar), wordt de waarde 1 teruggezonden.

Als u geen alias opgeeft, wordt uitgegaan van het actieve werkgebied.

Voorbeeld

U kunt het actieve recordnummer als volgt weergeven:

```
. USE Afnemers
. ? RECNO()
        1
. GO BOTTOM
        AFNEMERS: Recordnr. 8
. ? RECNO()
        8
. SKIP
        AFNEMERS: Recordnr. 9
. SELECT 2
. ? RECNO(1)
        9
. ? RECNO(Afnemers)
        9
```

Zie ook

SET RELATION

RECSIZE()

Met de functie RECSIZE() kunt u de grootte van een record in een actief database-bestand opvragen.

Syntaxis

RECSIZE([<alias>])

Gebruik

Als u geen alias opgeeft, wordt uitgegaan van het actieve werkgebied. Als er geen bestand actief is, wordt een nul teruggezonden door de functie RECSIZE().

Richtlijnen voor het programmeren

Combineer de functie RECSIZE() met de functies RECCOUNT() en DISKSPACE() in toepassingsprogramma's waarin automatisch een reservekopie wordt gemaakt van database-bestanden. U kunt met behulp van deze functies controleren of er op het station nog voldoende ruimte is voor het reservebestand. U moet dan ook weten hoe groot de aanhef van het bestand is. De grootte van de aanhef kan als volgt worden berekend:

32 * <aantal velden> + 34.

Voorbeeld

Met de volgende routine kunt u een groot database-bestand kopiëren van de vaste schijf naar een diskette in station B:

```
USE Bestnaam
Maanhef = (32 * <Aantal velden> + 35)
SET DEFAULT TO B   &&  B: is het doelstation.
SCAN
    WAIT "Plaats een nieuwe diskette in station B en druk op een toets..."
    IF DISKSPACE() - Maanhef <= 0
        ? "Niet genoeg ruimte op dat station."
        LOOP
    ENDIF
    COPY NEXT INT((DISKSPACE() - Maanhef)/RECSIZE()) TO Backup
    IF EOF()
        EXIT      && Geen records meer.
    ENDIF
ENDSCAN
CLOSE DATABASE
SET DEFAULT TO <Oorspronkelijk station>
```

Zie ook

DBF(), DISKSPACE(), RECCOUNT()

REPLICATE()

Met de functie REPLICATE() kunt u een tekenuitdrukking een bepaald aantal keren herhalen.

Syntaxis

REPLICATE(<Tuitdr>,<Nuitdr>)

Gebruik

Met de functie REPLICATE kunt u een tekenreeks maken door de tekenreeks in <Tuitdr> een bepaald aantal keren te herhalen. Het aantal keren dat de tekenreeks wordt herhaald, wordt bepaald door de waarde van <Nuitdr>.

De tekenreeks die wordt samengesteld, kan niet langer zijn dan 254 tekens. De numerieke uitdrukking moet dus een getal zijn dat kleiner is dan 254 gedeeld door het aantal tekens in <Tuitdr>.

Gebruik de functie REPLICATE() als een teken moet worden herhaald.

Voorbeeld

In het volgende voorbeeld wordt met behulp van de functie REPLICATE() een staafdiagram gemaakt van het veld Totaal_rek in het database-bestand Transact. Omdat de meeste waarden van het veld Totaal_rek groter zijn dan 254, wordt op schaal gewerkt, zodat het numerieke argument nooit groter is dan 50.

```
. SET HEADING OFF
. USE Transact
. CALCULATE MAX(Totaal_rek) TO grootste
12 records
     1850
. schaal = INT(grootste / 50 )
     37
. LIST Totaal_rek, REPLICATE("*", INT(Totaal_rek / schaal))
    1     1850.00  ***************************************************
    2     1200.00  *********************************
    3     1250.00  **********************************
    4     1250.00  **********************************
    5      415.00  ***********
    6      175.00  ****
    7     1000.00  ****************************
    8      700.00  *******************
    9      125.00  ***
   10     0450.00  ************
   11     1165.00  ****
   12    21500.00  ****************************************
```

RIGHT()

Met de functie RIGHT() kunt u een bepaald aantal tekens uit een tekenuitdrukking of memoveld lichten, te beginnen met het meest rechtse teken.

Syntaxis

RIGHT(<Tuitdr>/<naam memoveld>,<Nuitdr>)

Gebruik

Met de functie RIGHT() kunt u het laatste deel van een tekenreeks of memoveld opvragen. Met de numerieke uitdrukking geeft u op hoeveel tekens uit de tekenreeks of het memoveld moeten worden gelicht.

Als de numerieke uitdrukking nul of negatief is, wordt een lege tekenreeks teruggezonden.

Als de numerieke uitdrukking groter is dan de lengte van de tekenreeks, wordt de gehele tekenreeks teruggezonden. Het aantal tekens dat met de functie RIGHT() uit een tekenreeks of memoveld wordt teruggezonden, is maximaal 254.

Voorbeeld

U kunt als volgt de laatste drie tekens van een tekenuitdrukking opvragen:

```
. ? RIGHT("abcdef",3)
def
```

Zie ook

AT(), LEFT(), LTRIM(), RTRIM(), STUFF(), SUBSTR()

RLOCK()

Met de functie RLOCK() kunt u meerdere records vergrendelen. De functie is identiek aan LOCK().

Syntaxis

RLOCK(<[lijst Tuitdr>,<alias>]/[<alias>])

Gebruik

De tekenuitdrukking is een lijst met recordnummers (bijvoorbeeld "1,2,5,7,9"). Als u geen lijst met recordnummers of alias opgeeft, wordt het actieve record in het actieve database-bestand vergrendeld.

Als alias kunt u het nummer van een werkgebied of de aliasnaam van een database-bestand opgeven, of een indirecte verwijzing naar een bestand. Als u wel een alias opgeeft maar geen lijst met recordnummers, wordt het actieve record van het opgeven database-bestand vergrendeld.

Met deze functie vergrendelt u alle opgegeven records en alle actieve records die daaraan zijn gekoppeld. U kunt records koppelen met het commando SET RELATION. Er kunnen maximaal 50 records worden vergrendeld.

Wanneer een record wordt vergrendeld met RLOCK(), kunnen andere gebruikers de vergrendelde records nog steeds lezen, maar alleen de gebruiker die de records heeft vergrendeld, kan ze bewerken. Met deze functie kunt u een bepaald record of een groep gerelateerde records vergrendelen, zodat andere gebruikers nog steeds toegang hebben tot de niet vergrendelde records.

Als alle records in de lijst en alle gekoppelde records kunnen worden vergrendeld, resulteert de functie RLOCK() in de logische waarde waar (.T.). Als niet alle records kunnen worden vergrendeld, is de logische waarde onwaar (.F.) het resultaat en worden de records niet vergrendeld.

Een record dat u hebt vergrendeld met RLOCK() wordt pas ontgrendeld als u het commando UNLOCK geeft, het database-bestand sluit, of dBASE IV verlaat.

Voorbeeld

In het volgende voorbeeld wordt recordnr. 3 van het database-bestand Afnemers vergrendeld, terwijl Transact het actieve database-bestand is:

```
. CLEAR ALL
. USE Afnemers INDEX KIntnaam IN 2
. USE Transact
. ? RLOCK(3, "Afnemers")
.T.
```

Zie ook

FLOCK(), SET LOCK, SET RELATION, UNLOCK

ROLLBACK()

Met de functie ROLLBACK() kunt u bepalen of het laatste ROLLBACK-commando met succes is uitgevoerd.

Syntaxis

ROLLBACK()

Gebruik

U kunt de functie ROLLBACK() gebruiken om te controleren of een ROLLBACK-commando succesvol is afgesloten of niet.

De functie ROLLBACK() resulteert standaard in de logische waarde waar (.T.). Als het laatste ROLLBACK-commando om welke reden dan ook niet met succes is uitgevoerd (bijvoorbeeld een schijffout), blijft het resultaat onwaar (.F.) totdat er wel een ROLLBACK-commando met succes is uitgevoerd, of totdat u dBASE IV verlaat.

Voorbeeld

Zie het voorbeeld bij de functie COMPLETED() in dit hoofdstuk.

Zie ook

BEGIN/END TRANSACTION, COMPLETED(), ISMARKED(), ROLLBACK

ROUND()

Met de functie ROUND() kunt u getallen afronden op een bepaald aantal cijfers achter de komma.

Syntaxis

ROUND(<Nuitdr1>,<Nuitdr2>)

Gebruik

<Nuitdr1> is het getal of de numerieke uitdrukking die wordt afgerond. <Nuitdr2> is het aantal cijfers dat achter de komma mag blijven staan. Als <Nuitdr2> negatief is, wordt het getal afgerond op het opgegeven aantal cijfers voor de komma.

Voorbeelden

```
. ? ROUND(14.746321,2)
    14,75

. ? ROUND(17.321111,6)
    17,321000

. ? ROUND(10.7654321,0)
    11
. ? ROUND (14911,-3)
    15000
```

Negatieve getallen worden afgerond alsof ze positief zijn:

```
. ? ROUND(-5.8,0)
    -6

. ? ROUND(-5.2,0)
    -5
```

Zie ook

CEILING(), FLOOR(), INT(), STR(), VAL()

ROW()

Met de functie ROW() kunt u opvragen op welke rij de cursor zich bevindt.

Syntaxis

ROW()

Gebruik

Met de functie ROW() kunt u het beeldscherm relatief adresseren. Als er geen venster actief is en het commando vanaf de commandostip wordt gegeven, is de teruggezonden waarde 21 (SET STATUS ON) of 24 (SET STATUS OFF). Bij de beeldschermmodus met 43 lijnen zijn deze waarden respectievelijk 39 en 42.

Zie ook

@, COL(), PCOL(), PROW(),

RTOD()

Met de functie RTOD() kunt u radialen omzetten in graden.

Syntaxis

RTOD(<Nuitdr>)

De uitdrukking <Nuitdr> is een hoek gemeten in radialen.

Gebruik

Met de functie RTOD() kunt u radialen omzetten in graden.

Voorbeeld

In het volgende voorbeeld wordt 3Pi/2 radialen omgezet in graden:

```
. x = 3 * PI()/2
4,71
. ? RTOD(x)
270
```

Zie ook

ACOS(), ASIN(), ATAN(), ATN2(), COS(), DTOR(), SIN(), TAN()

RTRIM()

Met de functie RTRIM() kunt u alle volgspaties van een tekenreeks verwijderen. De functie werkt hetzelfde als de functie TRIM().

Syntaxis

RTRIM(<Tuitdr>) of TRIM(<Tuitdr>)

Gebruik

Met de functie RTRIM() kunt u volgspaties verwijderen uit tekenvelden. Wanneer SET SPACE op ON is ingesteld, wordt met TRIM(<veldnaam>,<veldnaam>) één spatie tussen de twee velden geplaatst. Met het commando TRIM[<veldnaam>) + <veldnaam> worden geen spaties geplaatst tussen de velden, ongeacht of SET SPACE op ON of op OFF is ingesteld.

Voorbeelden

In het volgende voorbeeld worden de velden Achternaam en Voornaam van het database-bestand Afnemers weergegeven:

```
. USE Afnemers
. ? Achternaam + Voornaam
Wijsman          Fred
```

De eerste naam wordt gescheiden van de tweede naam door de volgspaties van het veld achternaam. U kunt de volgspaties tussen de velden Achternaam en Voornaam als volgt verwijderen:

```
. ? TRIM(Achternaam) + Voornaam
WijsmanFred
```

In het volgende voorbeeld worden de volgspaties van de velden Achternaam en Voornaam verwijderd, maar de veldlengte blijft gelijk:

```
. DISPLAY TRIM(Achternaam), Woonplaats
Recordnr. TRIM(Achternaam)    Woonplaats
     1    Wijsman  Nieuw-Vennep

. DISPLAY TRIM(Achternaam)+", "+Voornaam, Woonplaats
Recordnr. TRIM(Achternaam)+", "+Voornaam  Woonplaats
     1    Wijsman, Fred  Nieuw-Vennep
```

Zie ook

LEFT(), LTRIM(), RIGHT(), SET SPACE

RUN()

De functie RUN() levert een code op om aan te geven dat een commando uit het besturingssysteem of een programma van dBASE IV is voltooid.

Syntaxis

RUN(<Tuitdr>)

In deze syntaxis is <Tuitdr> de naam van een commando, een batch-bestand of een ander programma dat door het besturingssysteem kan worden uitgevoerd.

Gebruik

Met RUN(<Tuitdr>) wordt het commando of programma uitgevoerd dat is opgegeven bij <Tuitdr>. U moet de bestandsextensie (indien van toepassing), de stationsaanduiding en het pad (als het programma niet op het standaard station staat en niet volgens het standaard pad toegankelijk is) opgeven. U kunt commandoregelargumenten opnemen achter de naam van het commando of programma.

In dBASE IV wordt de DOS COMSPEC-omgevingsvariabele gebruikt om de commando-interpreter te vinden. Om RUN() te kunnen gebruiken moet het bestand dat door COMSPEC wordt gespecificeerd, een bestaand bestand zijn. Als het bestand niet kan worden gevonden, verschijnt de melding Cannot load Command.com.

Uw computer moet over voldoende vrije geheugenruimte beschikken om de commando-interpreter en het uit te voeren programma te laden. Voordat het externe programma wordt uitgevoerd, wordt een deel van het geheugen dat door dBASE IV wordt gebruikt, vrijgemaakt.

Voer geen programma's uit die na beëindiging in het werkgeheugen blijven. Tot die TSR-programma's behoren ook enkele DOS-programma's zoals PRINT en ASSIGN. Deze programma's moet u uitvoeren voordat u dBASE IV start.

RUN() levert een numeriek argument op, maar dit argument is momenteel niet gedefinieerd.

Voorbeeld

In het volgende programma ziet u het commando om het programma Prog1.exe uit te voeren.

```
. RUN ("D:\Progs\Prog1.exe")
```

Zie ook

CALL, CALL(), LOAD, RUN/!

SEEK()

Met de functie SEEK() kunt u een bestand volgens de indexvolgorde doorzoeken.

Syntaxis

SEEK(<uitdr> [,<alias>])

Gebruik

Als <alias> kan het nummer van een werkgebied of de aliasnaam van een database-bestand worden opgegeven, of een indirecte verwijzing naar een geopend geïndexeerd database-bestand. <uitdr> is de uitdrukking waarnaar wordt gezocht. Als u geen alias opgeeft, wordt uitgegaan van het actieve werkgebied.

De uitdrukking die wordt opgegeven bij de functie SEEK(), wordt eerst verwerkt. Daarna wordt getracht de waarde van de uitdrukking te vinden in de hoofdindex van het database-bestand. Als de indexsleutel wordt gevonden, resulteert SEEK() in de logische waarde waar (.T.). Als de sleutel niet wordt gevonden, is de logische waarde onwaar (.F.) het resultaat.

Wanneer SET NEAR op ON is ingesteld, wordt de recordwijzer op het record geplaatst dat direct volgt op de mogelijke lokatie van de gezochte sleutel als geen exact overeenstemmende sleuteluitdrukking is gevonden.

Richtlijnen voor het programmeren

In een programma kunt u de functies SEEK() en FOUND() combineren om een bepaalde taak uit te voeren als het sleutelwoord wordt gevonden. Bijvoorbeeld:

```
SEEK M_nr
IF FOUND()
    * — Doe iets
ENDIF
```

Aangezien de functie SEEK() zowel de recordwijzer verplaatst als een logische waarde terugzendt, kunt u met deze functie de combinatie FIND/SEEK en FOUND() vervangen. Bijvoorbeeld:

```
IF SEEK(M_nr)
    * — Doe iets
ENDIF
```

Voorbeelden

U kunt als volgt de recordwijzer in een ander werkgebied verplaatsen:

```
. USE Afnemers INDEX Klntnaam
. SELECT 2
. USE Transact
. DISPLAY
     1    A10025   87-105    03-02-87 .T.   1850,00
. ? Afnemers->Klant_nr
A00001
. ? SEEK(Klant_nr, "Afnemers")
.T.
. ? Afnemers->Klant_nr
.
```

Met de functie SEEK() kunt u in een programma bepalen of er in een ander database-bestand een record voorkomt met hetzelfde sleutelveld, zonder een relatie te leggen met het commando SET RELATION. Bijvoorbeeld:

```
IF .NOT. SEEK(Klant_nr, "Afnemers")
     ? "Dit record komt niet voor in het database-bestand Afnemers."
ENDIF
```

Zie ook

EOF(), FIND, FOUND(), LOOKUP(), SEEK, SET NEAR

SELECT()

Met de functie SELECT() kunt u het nummer van het hoogste vrije werkgebied opvragen.

Syntaxis

SELECT()

Gebruik

Met de functie SELECT() kunt u het nummer van het hoogste vrije werkgebied opvragen. Er wordt een getal tussen 1 en 10 teruggezonden. Indien geen werkgebied meer vrij is, levert de functie de waarde 0 op.

Wanneer de werkgebieden 1 en 10 in gebruik zijn terwijl er ook een catalogus geopend is, wordt door SELECT() de waarde 9 teruggezonden.

Voorbeeld

```
. ? SELECT()
9
```

U kunt als volgt een bestand openen in een vrij werkgebied:

```
. bestnaam = "Afnemers"
. USE (Bestnaam) IN SELECT()
```

Zie ook

ALIAS()

Met de functie SET() kunt u de status van de verschillende SET-commando's opvragen.

Syntaxis

SET(<Tuitdr>)

Gebruik

Bij de tekenuitdrukking <Tuitdr> kunt u een sleutelwoord opgeven. In tabel 4-4 staan de sleutelwoorden die geldig zijn bij het SET-commando. Daarnaast worden de waarden vermeld die SET() oplevert bij het betreffende sleutelwoord.

Tabel 4-4 SET-sleutelwoorden en opgeleverde waarden

SET-sleutelwoord	Opgeleverde waarde	SET-sleutelwoord	Opgeleverde waarde
ALTERNATE	ON/OFF	DEVELOPMENT	ON/OFF
ATTRIBUTES	SET COLOR -waarden [1]	DIRECTORY	<pad>
AUTOSAVE	ON/OFF	DISPLAY	<weergavemodus>
BELL	ON/OFF	ECHO	ON/OFF
BLOCKSIZE	geheel getal	ENCRYPTION	ON/OFF
CARRY	ON/OFF	ESCAPE	ON/OFF
CATALOG	ON/OFF	EXACT	ON/OFF
CENTURY	ON/OFF	EXCLUSIVE	ON/OFF
CLOCK	ON/OFF	FIELDS	ON/OFF
COLOR	ON/OFF	FILTER	<filteruitdrukking>
CONFIRM	ON/OFF	FORMAT	<naam indelingsbestand>
CONSOLE	ON/OFF	FULLPATH	ON/OFF
CURSOR	ON/OFF	HEADINGS	ON/OFF
DEBUG	ON/OFF	HELP	ON/OFF
DECIMALS	geheel getal	HISTORY	ON/OFF
DEFAULT	<station>	HOURS	12 of 24
DELETED	ON/OFF	INSTRUCT	ON/OFF
DELIMITERS	ON/OFF	INTENSITY	ON/OFF
DESIGN	ON/OFF	LOCK	ON/OFF

(wordt vervolgd)

Tabel 4-4 SET-sleutelwoorden en opgeleverde waarden (vervolg)

SET-sleutelwoord	Opgeleverde waarde	SET-sleutelwoord	Opgeleverde waarde
MARGIN	geheel getal	SAFETY	ON/OFF
MEMOWIDTH	geheel getal	SCOREBOARD	ON/OFF
NEAR	ON/OFF	SPACE	ON/OFF
ODOMETER	geheel getal	SQL	ON/OFF
PATH	ON/OFF	STATUS	ON/OFF
PAUSE	ON/OFF	STEP	ON/OFF
PRECISION	geheel getal	TALK	ON/OFF
PRINTER	ON/OFF	TITLE	ON/OFF
PROCEDURE	<bestandnaam>	TRAP	ON/OFF
REFRESH	geheel getal	TYPEHEAD	geheel getal
RELATION	<uitdrukking> INTO	UNIQUE	ON/OFF
REPROCESS	geheel getal	VIEW	<bestandnaam>
	<alias>		

1) SET ATTRIBUTES levert de reeks attributen op zoals ingesteld bij SET COLOR. De attributen zijn NORMAL, HIGHLIGHT, beeldrand, vervolgens twee ampersands (&&) en daarna MESSAGES, TITLES, BOX, INFORMATION en FIELDS.

Voorbeelden

Om de status van SET CARRY vast te stellen typt u het volgende commando:

```
. ? SET ("CARRY")
OFF
```

Met het volgende programma kan de gebruiker de status van SET BELL wijzigen.

```
bellstat = IFF(SET("BELL") = "ON", "on.", "off.")
? "Bel is momenteel ingesteld op" + bellstat
ACCEPT "Wilt u de instelling wijzigen? (J/N) " TO wijzigen
IF wijzigen $ "Jj"
        SET BELL ON
        IF bellstat = "on"
                SET BELL OFF
        ENDIF
ENDIF
```

Met de functie SET() kunt u de status van de statusbalk onthouden:

```
statvlag = SET("STATUS")
        .
        .
        .
SET STATUS &statvlag.
```

U kunt de attributen opvragen zoals die bij SET COLOR zijn ingesteld.

```
. ? SET ("ATTRIBUTES")
W+/B, RG+/GB, N/N && W/N, W/B, RG+/GB, B/W, N/GB
```

Volgens deze instelling wordt de weergavegroep NORMAL helder wit op blauw HIGHLIGHT geel op cyaan, BEELDRAND zwart, MESSAGES wit op zwart (kan als helderwit op zwart worden weergegeven; zie tabel 3-4 bij het commando SET COLOR), TITLES wit op blauw (kan eveneens helder wit op blauw worden weergegeven), BOX geel op cyaan, INFORMATION blauw op wit en FIELDS zwart op cyaan.

Zie ook

LIST/DISPLAY STATUS, SET

Hoofdstuk 3, *SET-commando's*

SIGN()

Met de functie SIGN() kunt u het rekenkundige teken van een numerieke uitdrukking opvragen. Als de uitdrukking positief is, wordt de waarde 1 teruggezonden. Als de uitdrukking negatief is, wordt de waarde -1 teruggezonden. Als de uitdrukking 0 is, wordt de waarde 0 teruggezonden.

Syntaxis

SIGN(<Nuitdr>)

<Nuitdr> is een numerieke uitdrukking.

Gebruik

Met de functie SIGN() kunt u het teken van een numerieke uitdrukking bepalen, zonder de waarde van de uitdrukking te berekenen.

Voorbeeld

Gebruik de functie SIGN() als de uitkomst van een berekening hetzelfde teken moet hebben als de waarden van de berekening. In het volgende voorbeeld wordt het rekenkundige teken van het basisgetal opgeslagen in de variabele M_teken. Met deze variabele wordt uiteindelijk het teken van de uitkomst bepaald.

```
FUNCTION Macht
PARAMETERS M_grond, M_macht
* —Teken van grondgetal opslaan
m_teken = SIGN(M_grond)
M_tel = 1
DO WHILE M_tel < M_macht
    M_grond = M_grond * M_grond
    M_tel = M_tel + 1
ENDDO
RETURN(ABS(M_grond) * M_teken)
* EOP: Macht
```

Zie ook

ABS()

SIN()

Met de functie SIN() kunt u de sinus van een hoek berekenen.

Syntaxis

SIN(<Nuitdr>)

<Nuitdr> is een hoek gemeten in radialen.

Gebruik

Met de functie SIN() kunt u de sinus van een hoek berekenen. Er worden geen grenzen gesteld aan het argument. Het resultaat van de functie SIN() is een getal van het type F.

Voorbeeld

De waarde van de sinus ligt tussen +1 en 1. De sinus is nul bij een hoek van 0, Pi en 2Pi, gemeten in radialen.

```
. ? SI(PI()/2)
                1
. ? SIN(PI())
                0
. ? SIN(3*PI()/2)
                1
. ? SIN(2*PI())
                0
```

Met de commando's SET DECIMALS en SET PRECISION kunt u de nauwkeurigheid van de uitkomst bepalen.

Zie ook

ACOS(), ASIN(), ATAN(), ATN2(), COS(), DTOR(), PI(), RTOD(), TAN()

SOUNDEX()

Met de functie SOUNDEX() kunt u een tekenreeks fonetisch noteren. Op die manier kunt u een tekenreeks opzoeken als u niet zeker bent van de spelling.

Syntax

SOUNDEX(<Tuitdr>)

Gebruik

De functie SOUNDEX() zet de tekenreeks om in een code van vier tekens aan de hand van het volgend algoritme:

1. De eerste letter van de opgegeven uitdrukking <Tuitdr> wordt gehandhaafd.
2. De letters *a e h i o u w y* worden uit de tekenreeks verwijderd, met uitzondering van de eerste letter van de tekenreeks.
3. Aan de resterende letters wordt een cijfer toegekend:

   b f p v = 1
   c g j k q s x z = 2
   d t = 3
   l = 4
   m n = 5
   r = 6

4. Als twee of meer op elkaar volgende letters hetzelfde cijfer hebben, wordt alleen het eerste cijfer gehandhaafd.
5. De code die wordt teruggezonden heeft de indeling "letter cijfer cijfer cijfer". Als er niet genoeg cijfers zijn, wordt de code aangevuld met volgspaties. Als er meer dan drie cijfers zijn, vervallen alle cijfers rechts van het derde cijfer.
6. Als er een teken in de tekenreeks staat dat geen letter is, wordt alleen het deel van de tekenreeks omgezet dat voor dat teken staat.
7. Voorloopspaties worden overgeslagen.
8. Als het eerste teken van de tekenreeks geen letter is, wordt de code 0000 teruggezonden.

Deze acht stappen resulteren in een code van vier tekens. Met deze code kan worden gezocht naar tekenreeksen met een soortgelijk klankbeeld.

Voorbeeld

Met de functie SOUNDEX() kunt u namen met een bepaald klankbeeld opzoeken:

```
. USE Afnemers
. INDEX ON SOUNDEX(Voornaam) TO Alsnaam
100% geïndexeerd  8 records geïndexeerd
. ACCEPT "Geef op te zoeken naam op: " TO nieuwnaam
Geef op te zoeken naam op: Fleur
.
. nieuwnaam = SOUNDEX(Nieuwnaam)
F460
. SEEK nieuwnaam
. DISPLAY Voornaam
Recordnr.      Voornaam
     7         Floor
```

Zie ook

DIFFERENCE(), LOCATE

SPACE()

Met de functie SPACE() kunt u een tekenreeks maken die bestaat uit het opgegeven aantal spaties.

Syntaxis

SPACE(<Nuitdr>)

Gebruik

De tekenreeks die u maakt met SPACE() mag maximaal 254 spaties bevatten.

Voorbeeld

U kunt als volgt een geheugenvariabele maken die bestaat uit 20 spaties:

```
. STORE SPACE(20) TO spaties
. ? "*" + spaties + "*"
*                    *
```

SQRT()

Met de functie SQRT() kunt u de vierkantswortel van een positief getal berekenen.

Syntaxis

SQRT(<Nuitdr>)

Gebruik

Met de functie SQRT() kunt u de vierkantswortel berekenen van het getal dat is opgegeven bij <Nuitdr>. Het opgegeven getal kan zowel van het type N als van het type F zijn, maar de uitkomst is altijd een getal van het type F.

Met het commando SET DECIMALS kunt u instellen hoeveel decimalen worden weergegeven.

Voorbeeld

U kunt als volgt de vierkantswortel van 4 berekenen:

```
. ? SQRT(4)
          2
. ? SQRT(2*2)
          2
. SET DECIMALS TO 3
. STORE 4,000 TO getal
          4
. ? SQRT(getal)
          2
```

Zie ook

SET DECIMALS

STR()

Met de functie STR() kunt u een getal omzetten in een tekenreeks.

Syntaxis

STR(<Nuitdr> [,<lengte>] [,<decimalen>])

Standaardinstelling

Standaard is de lengte van de tekenreeks 10 tekens en wordt het getal afgerond op een geheel getal.

Gebruik

Bij de optie <lengte> kunt u opgegeven uit hoeveel tekens de tekenreeks moet bestaan, inclusief het decimaalteken, het minteken en het aantal decimalen.

Het getal dat u opgeeft bij de optie <decimalen>, bepaalt het totale aantal decimalen. Als het getal dat u opgeeft kleiner is dan het aantal cijfers rechts van het decimaalteken in de numerieke uitdrukking, worden asterisken teruggezonden in plaats van het getal.

Als u minder decimalen opgeeft dan er in de numerieke uitdrukking staan, wordt het getal afgerond op het opgegeven aantal decimalen.

Voorbeeld

U kunt het getal 11,14 * 10 als volgt weergeven als een tekenreeks:

```
. x = 11.14
                    11,14
. ? STR(x*10,5)
111
. ? STR(x*10,5,1)
111,4
. ? STR(x*10,5,2)
111,4
```

Zie ook

VAL()

STUFF()

Met de functie STUFF() kunt u een deel van een tekenreeks vervangen door een andere opgegeven tekenreeks.

Syntaxis

STUFF(<Tuitdr1>,<Nuitdr1>,<Nuitdr2>,<Tuitdr2>)

Gebruik

<Tuitdr1> kan een tekenuitdrukking of de naam van een variabele zijn. <Nuitdr1> en <Nuitdr2> zijn numerieke uitdrukkingen. <Nuitdr2> moet een nul of een positief getal zijn.

Met de functie STUFF() kunt u een deel van een tekenreeks wijzigen zonder de hele reeks opnieuw op te geven. De tekenreeks <Tuitdr2> wordt tussengevoegd in de tekenuitdrukking <Tuidr1>. Daarbij worden zoveel tekens vervangen als is aangegeven in <Nuitdr2>, te beginnen bij het teken dat op de positie staat die is aangegeven in <Nuitdr1>.

Als de beginpositie van de tekenreeks, aangegeven door <Nuitdr1>, gelijk is aan nul, wordt het door STUFF() beschouwd als 1. Als de beginpositie groter is dan de lengte van de tekenreeks uit <Tuitdr1>, wordt de tekenreeks <Tuitdr2> aan het einde van <Tuitdr1> geplakt.

Met <Nuitdr2> kunt u opgeven hoeveel tekens van de originele tekenreeks moeten worden verwijderd. Als nul wordt opgegeven, wordt de tweede tekenuitdrukking tussengevoegd en worden er geen tekens uit <Tuitdr1> verwijderd. Als het aantal tekens dat wordt opgegeven bij <Nuitdr2>, niet gelijk is aan het aantal tekens in <Tuitdr1>, is de nieuwe tekenreeks korter of langer dan de originele tekenreeks.

Wanneer <Tuitdr2> een nultekenreeks is, wordt het aantal tekens dat is opgegeven in <Nuitdr2> uit <Tuitdr1> verwijderd, zonder dat tekens worden toegevoegd.

WAARSCHUWING
De functie STUFF() wordt niet ondersteund in memovelden.

Voorbeeld

Met het volgende programma worden alle koppeltekens uit het veld Veld_1 vervangen door spaties:

```
SCAN
        DO WHILE "-" $ Veld_1
                REPLACE Veld_1 WITH STUFF(Veld_1, AT("-", Veld_1), 1, " ")
        ENDDO
ENDSCAN
```

Zie ook

LEFT(), RIGHT(), SUBSTR()

SUBSTR()

Met de functie SUBSTR() kunt u een bepaald aantal tekens uit een tekenuitdrukking of tekenvariabele lichten.

Syntaxis

SUBSTR(<Tuitdr>/<memoveld>,<beginpositie> [,<aantal tekens>])

Standaardinstelling

Als u niet opgeeft hoeveel tekens uit de tekenreeks moeten worden gelicht, wordt een subtekenreeks gemaakt die begint op de opgegeven beginpositie en doorloopt tot het einde van de originele tekenreeks.

Gebruik

Als <aantal tekens> groter is dan het aantal tekens dat tussen de beginpositie en het einde van de originele tekenuitdrukking staat, wordt een subtekenreeks teruggezonden die begint op de beginpositie en eindigt met het laatste teken van de originele tekenuitdrukking. De beginpositie moet een positief getal zijn.

Voorbeelden

U kunt als volgt de subtekenreeks "59" uit de tekenreeks "1958 1959 1960" lichten:

```
. ? SUBSTR("1958 1959 1960",8,2)
59
. STORE "Engels Spaans Italiaans Duits Frans" TO taal
Engels Spaans Italiaans Duits Frans
. ? SUBSTR(Taal,8,6)
Spaans
```

Zie ook

AT(), LEFT(), LTRIM(), RIGHT(), STR(), STUFF()

TAG()

Met de functie TAG() kunt u de naam opvragen van een label in een .mdx-bestand.

Syntaxis

TAG([<.mdx-bestand>,] <Nuitdr> [,<alias>])

Gebruik

Met de functie TAG() kunt u de naam opvragen van het indexbestand (.ndx) of de .mdx-label waarnaar wordt verwezen door <Nuitdr>. Als de opgegeven index niet bestaat, wordt een nultekenreeks teruggezonden. Als u geen .mdx-bestand opgeeft, wordt <Nuitdr> beschouwd als een verwijzing naar alle geopende indexen.

Wanneer u zowel een .ndx-bestand, een bijbehorend .ndx-bestand als een niet-bijbehorend .mdx-bestand hebt geopend, is de volgorde waarin de labels worden teruggezonden .ndx, bijbehorend .mdx en ten slotte niet-bijbehorend .mdx.

Als u geen aliasnaam opgeeft, wordt uitgegaan van het actieve werkgebied.

Voorbeeld

In het volgende voorbeeld wordt de tweede label van het bestand Afnemers.mdx weergegeven:

```
. USE Afnemers
. ? TAG(2)
KLANT_NR
```

Zie ook

DBF(), KEY(), MDX(), NDX(), ORDER()

TAN()

Met de functie TAN() kunt u de tangens van een hoek berekenen.

Syntaxis

TAN(<Nuitdr>)

Gebruik

De opgegeven numerieke uitdrukking is een hoek gemeten in radialen. Deze rekenkundige functie loopt van nul tot oneindig tussen 0 en Pi/2 radialen.

De teruggezonden waarde is een getal van het type F. Met de commando's SET DECIMALS en SET PRECISION kunt u de bepalen hoe nauwkeurig het getal wordt weergegeven.

Voorbeeld

```
. ?TAN(PI())
          0
```

Zie ook

ACOS(), ASIN(), ATAN(), ATN2(), COS(), SIN()

TIME()

Met de functie TIME() kunt u de systeemtijd opvragen in de vorm van een tekenreeks met de indeling uu:mm:ss.

Syntaxis

TIME()

Gebruik

U kunt de functie TIME() ook opnemen in berekeningen. Om dit te doen zet u de teruggezonden tekenreeks om in een numerieke waarde met de functies SUBSTR() en VAL().

Aangezien met de functie TIME() de tijd van het besturingssysteem wordt weergegeven, is het belangrijk dat u de systeemklok correct instelt. De functie werkt onafhankelijk van de instelling SET HOURS TO [12/24] en zal altijd een waarde terugzenden volgens de 24-uurs indeling.

Voorbeelden

```
. ? TIME()
21:22:23
. Mtijd = TIME()
21:22:23
. ? Mtijd
21:22:23

*Programma voor tijdsberekening
Begintijd = TIME()
Stand = 0
DO WHILE Stand < 100
                Stand = Stand + 1
ENDDO
Eindtijd = TIME ()
Oude_tijd = VAL(SUBSTR(Begintijd, 1, 2)) * 3600 +;
                VAL(SUBSTR(Begintijd, 4,2)) * 60 +;
                VAL(SUBSTR(Begintijd, 7,2))
Nieuw_tijd = VAL(SUBSTR(Eindtijd, 1, 2)) * 3600 +;
                VAL(SUBSTR(Eindtijd, 4,2)) * 60 +;
                VAL(SUBSTR(Eindtijd, 7,2))
Verschil = Nieuw_tijd Oude_tijd
? "Er zijn",LTRIM(STR(Verschil)), "seconden verstreken."
```

Zie ook

DATE()

TRANSFORM()

Met de functie TRANSFORM() kunt u tekengegevens, logische gegevens, datumgegevens en numerieke gegevens indelen zonder het commando @...SAY te gebruiken.

Syntaxis

TRANSFORM(<uitdr>,<Tuitdr>)

Standaardinstelling

De uitkomst van de functie TRANSFORM() is altijd een tekenuitdrukking, zelfs als de eerste uitdrukking van een ander type is.

Gebruik

Met de uitdrukking <Tuitdr> kunt u opgeven welke template moet worden gebruikt voor <uitdr>. <uitdr> kan in dat geval een tekenuitdrukking of een numerieke uitdrukking zijn.

Voor getallen en tekens kunt u bij de functie TRANSFORM() een template opgeven. Als u daarna het commando DISPLAY, LABEL, LIST of REPORT geeft, worden de gegevens overeenkomstig de template weergegeven.

Voorbeelden

Met de functie TRANSFORM kunt u het veld Achternaam van het database-bestand Afnemers zo indelen, dat spaties of koppeltekens worden weergeven tussen de letters:

```
. USE Afnemers
. ? TRANSFORM(Achternaam, "@R X X X X X X X")
W i j s m a n
. ? TRANSFORM(Achternaam, "@R X-X-X-X-X-X-X")
W-i-j-s-m-a-n
```

In het volgende voorbeeld wordt het veld Totaal_rek van het database-bestand Transact op twee verschillende manieren ingedeeld:

```
. USE Transact
. ? TRANSFORM(Totaal_rek, "@$ ##,###.##")
$1,850.00
. ? TRANSFORM(Totaal_rek, "@C")
1850,00 CR
```

Zie ook

@...PICTURE

TRIM()

De functie TRIM() is een alternatieve vorm van de functie RTRIM(). Raadpleeg het gedeelte over RTRIM() in dit hoofdstuk.

TYPE()

Met de functie TYPE() kunt u een tekenuitdrukking toetsen. De functie levert een hoofdletter op waarmee wordt aangeduid of het argument gegevens van het type teken, numeriek, logisch, memo, datum, zwevend of ongedefinieerd bevat.

Syntaxis

TYPE(<Tuitdr>)

Gebruik

De functie TYPE() zendt altijd een hoofdletter terug.

De hoofdletters die worden teruggezonden zijn:

Teken = C
Numeriek = N
Logisch = L
Memo = M
Datum = D
Zwevend = F
Ongedefinieerd = U

U kunt de functie TYPE() gebruiken om het voorkomen van een variabele of veld te controleren. Als de functie een U van ongedefinieerd oplevert, is het veld of de variabele onvindbaar.

Het argument bij TYPE() moet een tekenreeks tussen aanhalingstekens zijn. De tekenreeks tussen aanhalingstekens kan verwijzen naar de naam van een veld of een geheugenvariabele, maar moet in ieder geval betrekking hebben op een gegevensdeel. De functie levert de letter van het gegevenstype op van het gegevensdeel dat wordt genoemd in de tekenuitdrukking

Wanneer u een tekenreeks zonder aanhalingstekens gebruikt, wordt deze in dBASE IV behandeld als macrosubstitutie. Er wordt dan geprobeerd het argument te vergroten. Gebruik een argument zonder aanhalingstekens daarom alleen als dit daadwerkelijk kan worden vergroot tot een tekenuitdrukking. Na de vergroting levert TYPE() de waarde op die correspondeert met het gegevenstype uit de tekenuitdrukking.

Voorbeelden

Het argument van TYPE() is een tekenuitdrukking. U kunt echter ook een geheugenvariabele van het gegevenstype teken in het argument opgeven om te verwijzen naar variabelen van een andere type.

```
. tekenreeks = "Factuur"
Factuur
. ? TYPE("tekenreeks")      && tekenreeks is een tekenvariabele
C
. ? TYPE(tekenreeks)
U
. USE Transact
. ? TYPE(tekenreeks)        && Factuur is een veld in Transact.dbf.
L
```

Met de functie TYPE() kunt u bepalen of een uitdrukking geldig is of de foutmelding Gegevenstype onjuist zal opleveren. Om de uitdrukking "test/100" te testen en te bekijken of de uitdrukking geldig is, gebruikt u de volgende syntaxis:

```
. test = "A"
A
. ? TYPE("Test/100")
U
```

De functie levert hier een U op omdat een tekenreeks niet kan worden gedeeld door een getal.

```
. x = 5
                    5
. y = 7
                    7
. ? TYPE("x*y/4")
N
```

Hier levert TYPE() een N op omdat de bewerking een geldige uitdrukking oplevert.

Om de geldigheid van een uitdrukking te testen moet u de uitkomst van de functie vergelijken met de hoofdletter die het gegevenstype vertegenwoordigt.

```
. ? TYPE("(1+2)=5") = "L"
.T.
```

UPPER()

Met de functie UPPER() kunt u kleine letters omzetten in hoofdletters. De functie kan niet worden gebruikt bij een memoveld, tenzij eerst met de functie MLINE() een tekenreeks uit het memoveld is verwijderd.

Syntaxis

UPPER(<Tuitdr>)

Voorbeelden

```
. STORE "Dit is een mooie dag" TO mooiedag
Dit is een mooie dag
```

U kunt als volgt een variabele in hoofdletters weergeven:

```
. ? UPPER(Mooiedag)
DIT IS EEN MOOIE DAG
. ? mooiedag="DIT IS EEN MOOIE DAG"
.F.
. ? UPPER(niceday)="DIT IS EEN MOOIE DAG"
.T.
```

In het volgende voorbeeld wordt het eerste teken van het veld Re van een database-bestand omgezet in een hoofdletter:

```
. LIST RE
Recordnr.     RE
     1        an03
     2        an39
     3        an47
     4        an21
. REPLACE ALL Re WITH STUFF(Re, 1, 1, UPPER(SUBSTR(Re, 1, 1)))
4 records vervangen
. LIST RE
RECORDNR.     RE
     1        An03
     2        An39
     3        An47
     4        An21
```

Zie ook

ISALPHA(), ISLOWER(), ISUPPER(), LOWER(), MLINE()

USER()

Met de functie USER() kunt u de aanmeldingsnaam opvragen van de gebruiker die zich heeft aangemeld bij een beveiligd systeem.

Syntaxis

USER()

Gebruik

Wanneer in een systeem bestanden worden gecodeerd met PROTECT, kunt u met deze functie de aanmeldingsnaam opvragen van de gebruiker die zich heeft aangemeld. Als het systeem niet is beveiligd met PROTECT, wordt een nultekenreeks teruggezonden.

Zie ook

PROTECT

VAL()

Met de functie VAL() kunt u een getal dat is opgegeven in de vorm van een tekenreeks, omzetten in een numerieke uitdrukking.

Syntaxis

VAL(<Tuitdr>)

Gebruik

Als de opgegeven tekenuitdrukking begint met tekens die geen cijfers zijn, met uitzondering van spaties, wordt de waarde nul teruggezonden.

Met het commando SET DECIMALS kunt u instellen hoeveel decimalen worden weergegeven door de functie VAL().

De functie VAL() werkt van links naar rechts. Tekens worden omgezet in numerieke waarden totdat wordt geprobeerd een niet-numeriek teken om te zetten. Voorloopspaties worden genegeerd als er in het argument zowel numerieke als niet-numerieke tekens voorkomen. De numerieke tekens aan het begin van de tekenuitdrukking worden omgezet in een numerieke waarde. Volgspaties worden beschouwd als niet-numerieke tekens en de functie wordt beëindigd zodra wordt geprobeerd een volgspatie om te zetten.

De functie VAL() herkent alleen een punt (.) als decimaalteken. Een komma (,) wordt niet herkend als decimaalteken en ook niet als scheidingsteken voor duizendtallen.

Voorbeelden

Als een niet-numerieke tekenuitdrukking wordt opgegeven bij de functie VAL(), wordt de waarde nul teruggezonden:

```
. ?VAL("ABC")
0
. ?VAL("A=123")
0
. ?VAL("123=A")
123

. SET DECIMALS TO 1
. Y = VAL("123.45")
123,5
. ? STR(Y,6,2)
123,45
```

De waarde 123,5 wordt weergegeven door de functie VAL(), maar de waarde 123,45 wordt opgeslagen in het geheugen.

Zie ook

SET DECIMALS, STR()

VARREAD()

U kunt de functie VARREAD() gebruiken in een procedure die wordt aangeroepen door een ON-commando of een UDF tijdens het bewerken van een veld, een geheugenvariabele of een reekselement. De functie VARREAD() wordt gebruikt in combinatie met @-commando's en schermgrote bewerkcommando's om de naam te achterhalen van het element uit een reeks, het veld of de geheugenvariabele dat in bewerking is.

Syntaxis

VARREAD()

Gebruik

De functie VARREAD() levert het veld, de geheugenvariabele of het element uit een reeks op in hoofdletters. Aan de naam van een element uit een reeks worden ook de rij- en kolomcoördinaten toegevoegd. Als u een tekenreeks wilt vergelijken met een veldnaam die VARREAD() heeft opgeleverd, moet u de tekenreeks in hoofdletters intypen.

U kunt de functie gebruiken tijdens het bewerken van een veld. Wanneer u het commando READ geeft, worden tijdens de uitvoering van het commando de nog uit te voeren @...GET-commando's gewist.

Gebruik deze functie terwijl een veld wordt bewerkt. Als u het commando READ geeft, worden de niet uitgevoerde @...GET-commando's tijdens de uitvoering van het READ-commando gewist.

Voorbeeld

In de volgende procedure Mijnhulp is te zien hoe de functie VARREAD() kan worden gecombineerd met het commando WINDOW om speciale hulpschermen te maken. Als de gebruiker op **F1** drukt, wordt de procedure Mijnhulp uitgevoerd. In deze procedure wordt de toetsenbordbuffer gewist met de functie INKEY() omdat het commando ON KEY de buffer niet wist. Als de **F1**-toetsaanslag niet wordt gewist, wordt deze toetsaanslag gelezen door INKEY(0) als invoer van de gebruiker. Het hulpscherm zou dan direct weer verdwijnen, zonder dat wordt gewacht totdat de gebruiker op een toets drukt.

VARREAD()

```
ON KEY F1 DO Mijnhulp
@ 5,5 GET Achternaam
@ 6,5 GET Voornaam
READ

PROCEDURE Mijnhulp
null = INKEY()        && Toetsenbordbuffer wissen.
ACTIVATE WINDOW Hulpscherm
DO CASE
        CASE VARREAD() = "Achternaam"
                ? "Hulptekst voor Achternaam staat in Hulpscherm."
        CASE VARREAD() = "Voornaam"
                ? "Hulptekst voor Voornaam staat in Hulpscherm."
ENDCASE
        ? INKEY(0)        && Weergave vasthouden
DEACTIVATE WINDOW Hulpscherm
RETURN
* EOP: Mijnhulp
```

Zie ook

@, APPEND, BROWSE, EDIT, FIELD(), READ

VERSION()

Met de functie VERSION() kunt u het versienummer van dBASE IV opvragen.

Syntaxis

VERSION()

Gebruik

Met de functie VERSION() kunt u het versienummer van het programma opvragen. Het versienummer wordt ook weergegeven als u dBASE IV start.

Deze functie wordt vooral gebruikt in programma's waarin functies zijn opgenomen die specifiek zijn voor een bepaalde versie.

Voorbeelden

U kunt het versienummer als volgt opvragen:

```
. ? VERSION()
dBASE IV 1.0
```

Zie ook

OS()

WINDOW()

Met de functie WINDOW() kunt u de naam opvragen van het actieve venster.

Syntaxis

WINDOW()

Gebruik

De functie WINDOW() levert de naam van het actieve venster op in de vorm van een tekenreeks. Als er geen venster actief is, levert de functie een nultekenreeks op.

Er kunnen weliswaar meerdere venster tegelijk op een scherm worden weergegeven, maar slechts een venster kan actief zijn. Indien u met ACTIVATE WINDOW een lijst met vensters actief zou maken, is alleen het laatste venster van de lijst actief.

Voorbeeld

U kunt de naam van het actieve venster opslaan en het venster later opnieuw activeren.

```
. DEFINE WINDOW Invoer1 FROM 2,4 TO 15,30 DOUBLE
. DEFINE WINDOW Invoer2      FROM 5,5 TO 20,20 DOUBLE
. ACTIVATE WINDOW Invoer1
. Wactief = WINDOW()
. ACTIVATE WINDOW Invoer2
.
.
.
. ACTIVATE WINDOW &Wactief
```

Zie ook

ACTIVATE WINDOW, DEFINE WINDOW

YEAR()

Met de functie YEAR() kunt u het jaar van een datumuitdrukking opvragen. Het jaar wordt altijd teruggezonden in de vorm van een numerieke uitdrukking van vier cijfers.

Syntaxis

YEAR (<Duitdr>)

Gebruik

Met de functie YEAR() kunt u indexeren op een datumveld in combinatie met een tekenveld, bijvoorbeeld een naamveld.

Voorbeelden

Gesteld dat 15-05-88 de systeemdatum is:

```
. ? YEAR(DATE())
                    1988
```

U kunt het jaar van de systeemdatum ook opslaan in een geheugenvariabele:

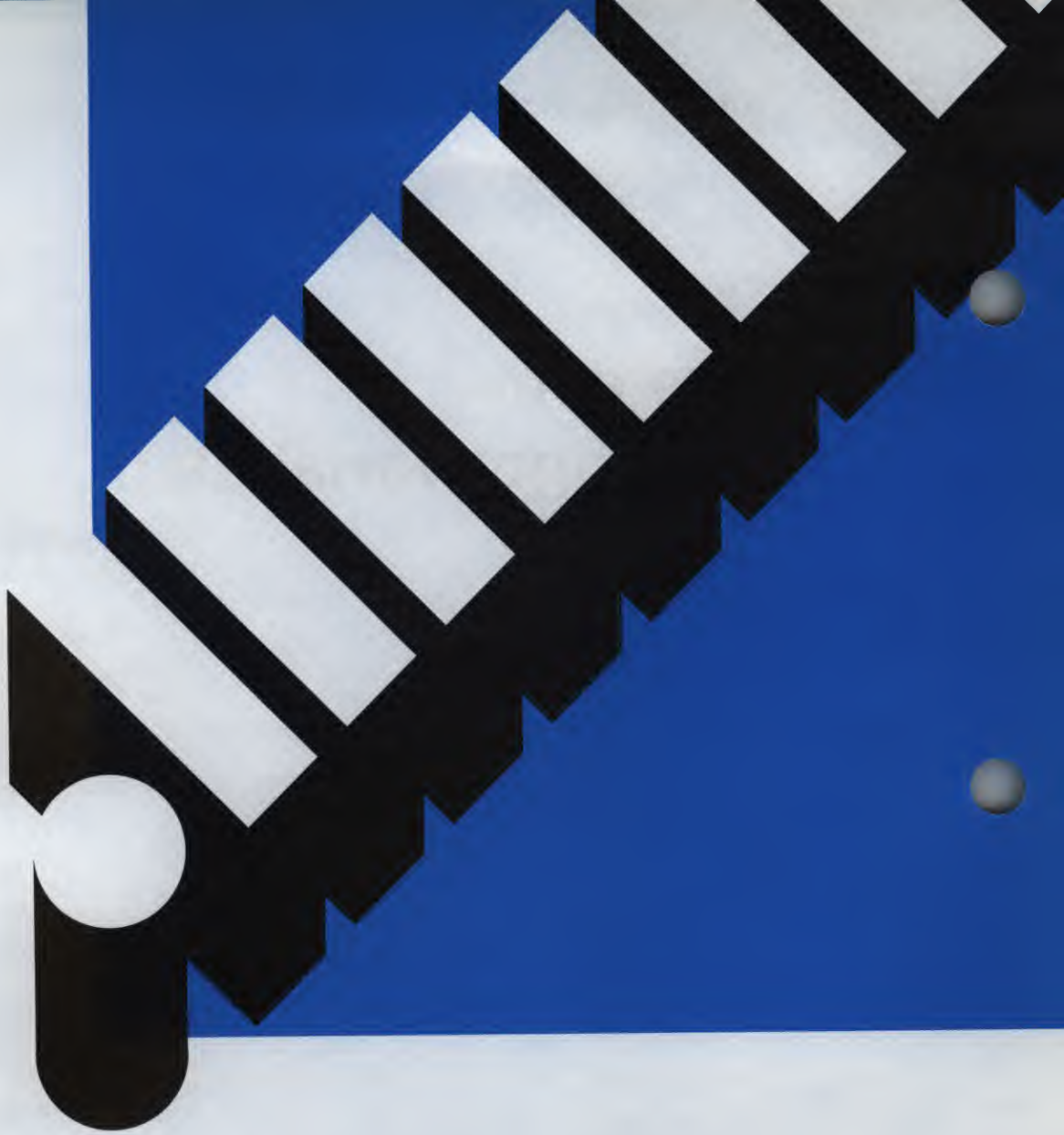
```
. Mjaar = YEAR(DATE())
                    1988

. ? TYPE(Mjaar)
N
```

In het volgende voorbeeld wordt 50 jaar opgeteld bij het jaar van de systeemdatum en wordt het aldus verkregen jaartal opgeslagen in de numerieke variabele Nieuwjaar:

```
. Nieuwjaar = YEAR(DATE()) + 50
                    2038
```

Naslagshandboek van dBASE IV

Systeem-geheugenvariabelen



Systeemgeheugen-variabelen

5

_alignment

Met _alignment wordt bepaald hoe de uitvoer van het commando ?/?? wordt uitgelijnd ten opzichte van de marges, als _wrap op waar (.T.) is ingesteld.

Syntaxis

_alignment = "LEFT"/"center"/"right"

Standaardinstelling

Standaard wordt de uitvoer van het commando ?/?? links uitgelijnd.

Gebruik

Met _alignment kunt u bepalen of de uitvoer van de commando's ? en ?? tegen de linker marge wordt uitgelijnd, tegen de rechter marge wordt uitgelijnd, of wordt gecentreerd tussen de marges. U kunt de uitlijning zo vaak wijzigen als u wilt.

Met deze systeemvariabele wordt de tekst uitgelijnd *tussen de marges*. Verwar deze systeemvariabele niet met uitlijnfuncties van PICTURE (B, I en J). Deze uitlijnfuncties bepalen namelijk hoe tekst wordt uitgelijnd *binnen templates*.

_alignment heeft geen invloed als de systeemvariabele _wrap niet op waar (.T.) is ingesteld.

Opties

"LEFT" — Tekst wordt uitgelijnd langs de linker marge (links gelijnd).

"CENTER" — Tekst wordt gecentreerd tussen de linker en rechter marge.

"RIGHT" — Tekst wordt uitgelijnd langs de rechter marge (rechts gelijnd).

Voorbeeld

In de volgende procedure worden de datum en het paginanummer aan de rechter kant van de pagina afgedrukt. Met het commando ?? wordt het paginanummer op dezelfde regel als de datum afgedrukt.

_alignment

```
_wrap = .T.
SET PRINTER ON
SET TALK OFF
_alignment = RIGHT
?? "Pagina", STR(_pageno, 4, 0)," Datum:", DATE()
SET PRINTER OFF
_alignment = "LEFT"
```

Afhankelijk van de waarden van DATE() en _pageno wordt de informatie als volgt afgedrukt:

```
                                        Pagina    1          Datum:    29-07-88
```

Zie ook

_lmargin, _rmargin, _wrap, @ (B, I en J van PICTURE), STR()

_box

Met _box wordt bepaald of kaders die zijn gedefinieerd met het commando DEFINE BOX worden weergegeven.

Syntaxis

_box = <voorwaarde>

Standaardinstelling

Standaard is _box ingesteld op waar (.T.). Dat betekent dat de kaders worden weergegeven.

Gebruik

Met de systeemvariabele _box kunt u bepalen of de kaders die u hebt gedefinieerd met het commando DEFINE BOX worden weergegeven. Als u _box instelt op waar (.T.), worden de kaders weergegeven. Als _box op onwaar (.F.) is ingesteld, worden de kaders niet weergegeven.

U kunt een kader tekenen met het commando @...TO, of _box nu op waar (.T.) of onwaar (.F.) is ingesteld.

Met _box kunt u in een programma het exacte moment bepalen waarop een kader of deel van een kader wordt weergegeven. U kunt bijvoorbeeld een kader gedeeltelijk afdrukken door _box in te stellen op onwaar (.F.) voordat het kader geheel is afgedrukt. Als u later _box weer op waar (.T.) instelt, wordt de rest van het kader afgedrukt.

Voorbeeld

In het volgende programma worden met de functie SPACE() 51 tekens overschreven op de bovenste en onderste regel van het kader, zodat alleen de hoeken worden afgedrukt. Het doel van de toewijzingsopdracht na het commando SCAN is ervoor te zorgen dat het kader alleen wordt weergegeven (_box = .T.) voor de eerste en laatste drie records.

```
*Box.prg
SET TALK OFF
USE Magazijn ORDER Artikel
DEFINE BOX FROM 10 TO 70 HEIGHT 8 DOUBLE
_box = .T.
?? SPACE(51) AT 15
?
Tel = 1
SCAN
        _box = (Tel < 4 .OR. Tel > (RECCOUNT() - 3))
        ?? Artikel AT 12, Omschrijving
        ?
        Tel = Tel + 1
ENDSCAN
?? SPACE(51) AT 15
?
```

De uitvoer van het programma ziet er als volgt uit:

```
BOEKENKAST                TEAKHOUT,2 PLANKEN
BUREAU,DIRECTIE,5 POTEN   EIKENHOUT,FINEER
DOSSIERKAST,2 LADEN       METAAL,BRUIN
DOSSIERKAST,4 LADEN       METAAL,BRUIN
LADENKAST,2 LADEN         METAAL,ZWART
LAMP,STAAND               MESSING,6 POTEN,ENGELS
LAMP,STAAND               MESSING,6 POTEN,ART DECO
LAMP,STAAND               MESSING,6 POTEN,ENGELS
SOFA,6 POTEN              LEDER,BRUIN,HOGE RUGLEUNING
SOFA,6 POTEN              FLUWEEL,GRIJS,FRANS
SOFA,8 POTEN              FLUWEEL,BLAUW,FRANS
STOEL,BUREAU              LEDER,BRUIN,HOGE RUGLEUNING
STOEL,BUREAU              LEDER,BRUIN,HOGE RUGLEUNING
STOEL,BUREAU              LEDER,BRUIN
STOEL,BUREAU              LEDER,BRUIN
STOEL,KANTINE             PLASTIC,GRIJS
TAFEL,BIJZET              EIKENHOUT,2 POTEN,VIERKANT
```

Afbeelding 5-1 Uitvoer in een kader

DEFINE BOX, SPACE()

_indent

Met _indent wordt bepaald hoeveel de eerste regel inspringt van iedere nieuwe alinea die wordt weergegeven met het commando ?. Deze systeemvariabele heeft alleen invloed als _wrap op waar (.T.) is ingesteld.

Syntaxis

_indent = <Nuitdr>

Standaardinstelling

De standaardinstelling van _indent is 0. Een nieuwe alinea springt dan niet in.

Gebruik

Met de systeemvariabele _indent kunt u opgeven hoeveel tekens de eerste regel van elke nieuwe alinea moet inspringen. _indent heeft een bereik van -(_lmargin), de negatieve waarde van de systeemvariabele _lmargin, tot en met (_rmargin - _lmargin - 1). Stel dat _lmargin is ingesteld op 10, dan kan _indent worden ingesteld op -10. De eerste regel van iedere alinea begint dan tien kolommen links van de linker marge. Als _rmargin op 75 is ingesteld, kan _indent maximaal op 64 worden ingesteld.

Iedere keer dat het commando ? wordt gegeven, wordt begonnen met een nieuwe alinea. Als het commando ?? wordt gegeven, wordt verder gegaan met de actieve alinea.

De som van _indent en _lmargin (_indent + _lmargin) moet kleiner zijn dan de waarde van _rmargin. Als _wrap niet op waar (.T.) is ingesteld, heeft _indent geen invloed.

Wanneer de uitvoer naar de printer wordt gezonden, wordt bij het instellen van de marges rekening gehouden met de instelling van _ploffset.

OPMERKING
In afbeelding 1-1 in hoofdstuk 1, "Basisbegrippen", wordt de instelling van _indent weergegeven.

Zie ook

_lmargin, _ploffset, _rmargin, _wrap, ?/??

_lmargin

Met _lmargin wordt de linker marge ingesteld voor de uitvoer van het commando ?. _lmargin is alleen actief als _wrap op waar (.T.) is ingesteld.

Syntaxis

_lmargin = <Nuitdr>

Standaardinstelling

De standaardinstelling van _lmargin is 0.

Gebruik

De linker marge is het aantal tekens dat wordt verzonden voor elke niet inspringende regel van een alinea. De waarde die aan de systeemvariabele _lmargin kan worden toegewezen varieert 0 tot en met 254. _lmargin heeft geen invloed, tenzij _wrap op waar (.T.) is ingesteld.

Er is verschil tussen _lmargin en _ploffset. _ploffset is het aantal tekens tussen de linker zijde van het papier en de positie waarop wordt begonnen met afdrukken als de linker marge is ingesteld op nul. De linker marge is het aantal tekens tussen de eerst mogelijke kolom waarin kan worden afgedrukt en de kolom waarin werkelijk wordt afgedrukt als een regel niet inspringt. _ploffset heeft alleen invloed op de uitvoer die naar de printer wordt gezonden.

OPMERKING
In afbeelding 1-1 in hoofdstuk 1, "Basisbegrippen" wordt de instelling van _lmargin weergegeven.

De som van _lmargin en _indent (_lmargin + _indent) moet kleiner zijn dan de waarde van _rmargin.

Voorbeeld

In het volgende programmabestand wordt _lmargin gebruikt om een lijst te laten inspringen ten opzichte van de titel:

```
SET TALK OFF
USE Afnemers ORDER Klant_nr
_wrap = .T.
_lmargin = 0
? "Lijst met huidige klanten"
?
_lmargin = 10
SCAN
      ? Klant_nr + "    ", Klantnaam
ENDSCAN
```

De uitvoer van het programma ziet er als volgt uit:

```
Lijst met huidige klanten

    A00001      WIJSMAN & ZONEN BV
    A00005      SMITS & SMITS
    A10025      PUNTER PROMOTIONS
    B12000      VOLLEBREGT IMPORT
    C00001      L. G. BLOEM & PARTNERS
    C00002      TIEMESE & KEEFMAN BV
    L00001      BALJUW & CO
    L00002      SAUER & LANGWEER
```

Zie ook

_indent, _ploffset, _rmargin, _wrap, ?/??

_padvance

Met _padvance wordt bepaald hoe de printer het papier doorvoert.

Syntaxis

_padvance = “FORMFEED”/”linefeeds”

Standaardinstelling

De standaardinstelling van _padvance is “FORMFEED”.

Gebruik

Met _padvance kunt u bepalen of het papier wordt doorgevoerd met regeldoorvoer (line feeds) of paginadoorvoer (form feeds).

Wanneer het papier wordt doorgevoerd met paginadoorvoer, gebeurt dat volgens de op de printer ingestelde lengte van het papier.

Wanneer het papier wordt doorgevoerd met regeldoorvoer, wordt eerst bepaald of een paginadoorvoerteken is verzonden in de uitvoerstroom voordat wordt berekend hoeveel regels er nodig zijn om de volgende pagina te bereiken.

Als een paginadoorvoerteken is verzonden in de uitvoerstroom, wordt aan de hand van de formule (_plength - _plineno) berekend hoeveel regels moeten worden verzonden. Dit gebeurt:

- wanneer u het commando EJECT PAGE geeft zonder de optie ON PAGE;
- wanneer u het commando EJECT PAGE geeft met de optie ON PAGE en de waarde van de actieve regelpositie hoger is dan die van de regel die is opgegeven bij ON PAGE;
- wanneer in het programma het commando PRINTJOB of ENDPRINTJOB wordt uitgevoerd en de pagina afhankelijk van de instelling van de systeemgeheugenvariabele _peject wordt doorgevoerd.

Als het paginadoorvoerteken niet wordt verzonden in de uitvoerstroom, wordt de formule (_plength MOD(PROW(), _plength)) gehanteerd. Dat gebeurt bijvoorbeeld als u het commando EJECT geeft, of als u SET DEVICE instelt op PRINTER en een paginadoorvoerteken verzendt met het commando @.

Als u korte pagina’s wilt afdrukken, bijvoorbeeld cheques die 20 regels lang zijn, hoeft u de op de printer ingestelde lengte van het papier niet te wijzigen. In plaats daarvan stelt u _plength in op het gewenste aantal regels en _padvance op “LINEFEEDS”. De op de printer ingesteld lengte van het papier doet dan niet terzake, omdat er geen paginadoorvoertekens worden verzonden.

Opties

"FORMFEED" — Het papier wordt per vel getransporteerd.

"LINEFEEDS" — Het papier wordt per regel getransporteerd.

Voorbeeld

In het volgende voorbeeld wordt met het programma Chq_afdr.prg een cheque afgedrukt voor ieder record in het database-bestand Debtr.dbf. Het programma Chq_afdr.prg staat niet op de diskettes van dBASE IV.

```
PRIVATE _plength, _padvance
_plength = 20
_padvance = "LINEFEEDS"
USE Debtr
SCAN
    DO Chq_afdr        && Druk een cheque af.
    EJECT              && Ga naar het begin van de volgende cheque.
ENDSCAN
CLOSE DATABASE
```

Zie ook

_peject, _plength, _plineno, EJECT, PROW()

_pageno

Met _pageno kunt u het actieve paginanummer bepalen of instellen.

Syntaxis

_pageno = <Nuitdr>

Standaardinstelling

Het standaard paginanummer is 1.

Gebruik

_pageno werkt op de gegevensstroom die wordt uitgevoerd. Deze *uitvoerstroom* wordt geproduceerd door alle commando's die gegevens uitvoeren, met uitzondering van de commando's @, @...TO en EJECT.

Met de systeemvariabele _pageno kunt u zowel het actieve paginanummer bepalen als het paginanummer instellen op een bepaalde waarde. Met deze variabele kunt u paginanummers afdrukken in een rapport of, wanneer u documenten combineert, het juiste paginanummer toewijzen aan de eerste pagina van het tweede document. Aan de systeemvariabele _pageno kan een geheel getal worden toegewezen van 1 tot en met 32767.

Het paginanummer van de actieve uitvoerstroom wordt automatisch bijgehouden. De variabele wordt met één verhoogd als het begin van de volgende pagina wordt bereikt, omdat het punt waarop een pagina wordt afgebroken afhankelijk is van de tekst in het rapport. De waarde van _pageno is afhankelijk van de waarde van _plineno en _plength. De systeemvariabelen _plineno en _plength worden elders in dit hoofdstuk behandeld.

Verwar _pageno niet met _pbpage. _pbpage is de systeemvariabele waarmee wordt bepaald op welke pagina moet worden begonnen met afdrukken.

Tip

De drie systeemvariabelen _pageno, _pbpage en _pepage moeten met elkaar in overeenstemming zijn. Wanneer bijvoorbeeld _pbpage = 3, _pepage = 5 en _pageno = 10, wordt er geen enkele pagina afgedrukt, omdat het actieve paginanummer buiten het bereik van de af te drukken pagina's ligt (3 tot en met 5).

De waarde van _pageno moet kleiner dan of gelijk zijn aan de waarde van _pepage. In het vorige voorbeeld moet _pageno kleiner dan of gelijk zijn aan 5, anders worden er geen pagina's afgedrukt. Door _pageno, _pbpage en _pepage op de juiste waarden in te stellen, kunt u een bepaald deel van een document afdrukken.

Bij REPORT FORM en LABEL FORM wordt het paginanummer altijd op opnieuw ingesteld. Als u een ander paginanummer wilt geven aan de eerste pagina van uw rapport, moet u naar het menu **Afdrukken** gaan en daar de waarde van de optie **Nummer eerste pagina** wijzigen.

Voorbeelden

In de volgende procedure wordt het paginanummer gecentreerd afgedrukt tussen streepjes, onder aan elke pagina van het rapport. Met de commando's ON PAGE en DO kunt u instellen dat deze procedure wordt uitgevoerd als de printkop op een bepaalde regel van de pagina staat.

```
PROCEDURE Voettkst
PRIVATE _wrap, _alignment
_wrap = .T.
_alignment = "CENTER"
? "-", _pageno PICTURE "@T 9999", "-"
?
RETURN
```

In de volgende procedure worden de tweede en derde fysieke pagina van een document afgedrukt met de paginanummers 5 en 6.

```
_plineno = 0
_pageno = 5
_pbpage = 6          && tweede fysieke pagina
_pepage = 7          && derde fysieke pagina
PRINTJOB
.
.
.
```

Zie ook

_pbpage, _pepage, _plength, _plineno, ON PAGE

_pbpage

Met _pbpage wordt bepaald welke pagina als eerste moet worden afgedrukt.

Syntaxis

_pbpage = <Nuitdr>

Standaardinstelling

Standaard wordt op pagina 1 begonnen met afdrukken.

Gebruik

Met de systeemvariabele _pbpage kunt u opgeven dat alle pagina's met een lager paginanummer dan het opgegeven paginanummer niet worden afgedrukt. Alle pagina's voor de opgegeven pagina worden niet afgedrukt.

Het paginanummer dat kan worden opgegeven, varieert van 1 tot en met 37767. De waarde van _pbpage moet kleiner dan of gelijk zijn aan de waarde van _pepage (zie de beschrijving elders in dit hoofdstuk).

Wanneer gegevens worden uitgevoerd, wordt voortdurend gecontroleerd of het actieve paginanummer (_pageno) groter dan of gelijk is aan het nummer van de beginpagina (_pbpage). Als _pageno kleiner is dan _pbpage, wordt de uitvoer alleen intern verschoven om de waarde van _pcolno, _plineno en _pageno te verhogen. Er wordt geen uitvoer naar het scherm of de printer gezonden totdat _pbpage is bereikt.

Tip

De drie systeemvariabelen _pageno, _pbpage en _pepage moeten met elkaar in overeenstemming zijn. Wanneer bijvoorbeeld _pbpage = 3, _pepage = 5 en _pageno = 10, wordt er geen enkele pagina afgedrukt, omdat het actieve paginanummer buiten het bereik van de af te drukken pagina's ligt (3 tot en met 5).

De waarde van _pageno moet kleiner dan of gelijk zijn aan de waarde van _pepage. In het vorige voorbeeld moet _pageno kleiner dan of gelijk zijn aan 5, anders worden er geen pagina's afgedrukt. Door _pageno, _pbpage en _pepage op de juiste waarden in te stellen, kunt u een bepaald deel van een document afdrukken.

Het actieve paginanummer kan overeenstemmen met de pagina die wordt afgedrukt, maar niet noodzakelijkerwijs. _pageno is immers een systeemgeheugenvariabele waaraan een waarde kan zijn toegekend.

Voorbeeld

Met de systeemvariabele _pbpage kunnen programmeurs gebruikers in staat stellen de uitvoer op een bepaalde pagina te laten beginnen. Op die manier kan tijd worden bespaard wanneer zich halverwege het afdrukken van een groot document een fout voordoet bij de printer, bijvoorbeeld papier dat vastloopt. Stel dat het afdrukken van het document Lang_rpt is onderbroken. U kunt dan op de volgende manier de resterende pagina's van het rapport afdrukken:

```
pbegin = 1
@ 10,20 SAY "Begin op pagina" GET pbegin PICTURE "99999"
READ
_pbpage = pbegin
DO Lang_rpt
```

Zie ook

_pageno, _pepage, PRINTJOB/ENDPRINTJOB

_pcolno

Met _pcolno kunt u het actieve kolomnummer opvragen of de uitvoerstroom in de opgegeven kolom van de actieve regel laten beginnen. Een toewijzing aan _pcolno is gelijkwaardig aan de AT-clausule van het commando ?/??.

Syntaxis

_pcolno = <Nuitdr>

Gebruik

Met _pcolno kunt u de uitvoerpositie verplaatsen naar een bepaalde kolom op het scherm, op de printer of in het bestand waarnaar de uitvoer wordt gezonden. U kunt aan _pcolno een geheel getal toewijzen van 0 tot en met 255.

Alle commando's die gegevens uitvoeren produceren een *uitvoerstroom*, met uitzondering van de commando's @, @...TO en EJECT. De bestemming van de uitvoerstroom kan worden bepaald met de commando's SET CONSOLE, SET PRINTER en SET ALTERNATE, of met de opties TO PRINTER/TO FILE <bestandsnaam> van commando's als LIST/DISPLAY.

OPMERKING
De uitvoerstroom moet niet worden gecombineerd met uitvoer van het commando @...SAY.

Als _wrap op onwaar (.F.) en SET PRINTER op ON is ingesteld, kunt u eerder uitgevoerde tekst overschrijven door de waarde van _pcolno te verlagen.

Als _wrap op waar (.T.) is ingesteld en u de waarde van _pcolno verlaagt, wordt de tekst in de interne buffer overschreven en niet afgedrukt.

Met de functie PCOL() kunt u de actieve positie van de printkop opvragen. Als SET PRINTER op OFF is ingesteld, verandert de waarde van PCOL() niet. Met _pcolno kunt u echter de actieve positie van de uitvoerstroom instellen of opvragen. De waarde van _pcolno verandert zolang gegevens worden uitgevoerd, ongeacht de instelling van SET PRINTER.

U kunt de waarde van _pcolno ook wijzigen met de clausule AT van het commando ?/??.

Voorbeeld

In het volgende voorbeeld wordt het woord "veranderd" doorgehaald met schuine strepen (/). Hiertoe wordt de printkop op de eerste letter van het woord "veranderd" geplaatst, waarna elk teken van het woord wordt doorgehaald met een schuine streep.

```
_wrap = .F.
? "Deze tekst is veranderd"
_pcolno = _pcolno LEN("veranderd")
?? REPLICATE(/, LEN("veranderd"))
?? "gewijzigd."
?
```

De uitvoer ziet er als volgt uit:

```
Deze tekst is v/e/r/a/n/d/e/r/d/ gewijzigd.
```

Zie ook

_plineno, _rmargin, LEN(), PCOL(), REPLICATE(), SET PRINTER, TRIM()

_pcopies

Met _pcopies kunt u opgeven hoeveel exemplaren van een document moeten worden afgedrukt.

Syntaxis

_pcopies = <Nuitdr>

Standaardinstelling

Standaard wordt slechts één exemplaar afgedrukt.

Gebruik

Met de systeemvariabele _pcopies kunt u bepalen hoeveel exemplaren van een document worden afgedrukt. De waarde die kan worden opgegeven is een geheel getal tussen 1 en 32767.

U kunt _pcopies alleen gebruiken in programma's, omdat de systeemvariabele moet worden gecombineerd met het commando PRINTJOB/ENDPRINTJOB. De systeemvariabele _pcopies moet zijn gedefinieerd voordat het commando PRINTJOB wordt uitgevoerd.

OPMERKING
Als u _pcopies op een waarde hebt ingesteld die groter dan 1 is, wordt de uitvoer met het commando PRINTJOB naar uw vaste schijf gezonden. Van daaruit wordt de uitvoer zo vaak als opgegeven in _pcopies naar de printer gezonden. Vandaar dat uitdrukkingen in het commando PRINTJOB maar eenmaal worden geëvalueerd worden. In het volgende voorbeeld worden vijf exemplaren gemaakt van "Rapport Kopie".

```
_pcopies = 5
PRINTJOB
    ? "Rapport Kopie", STR(x,1)
    x=x+1
    *Hier is het restant van het rapport
ENDPRINTJOB
```

Voorbeeld

In het volgende programma kan de gebruiker kiezen hoeveel exemplaren van een rapport moeten worden afgedrukt (raadpleeg het gedeelte over het commando PRINTJOB/ENDPRINTJOB voor meer informatie over het afdrukken van gegevens in programma's).

```
exemplaren = 1
@ 10,20 SAY "Aantal exemplaren: " GET exemplaren PICTURE "99"
READ
_pcopies = exemplaren
PRINTJOB
        . && <commando's>
        .
        .
ENDPRINTJOB
```

Zie ook

@, PRINTJOB/ENDPRINTJOB

_pdriver

Met _pdriver kunt u de naam van het actieve printerstuurprogramma opvragen of een bepaald printerstuurprogramma activeren.

Syntaxis

_pdriver = ''<naam printerstuurprogr>''

Standaardinstelling

Wanneer u dBASE IV installeert kiest u een printerstuurprogramma. Dit standaard printerstuurprogramma wordt opgenomen in het bestand Config.db. Raadpleeg *Aan de slag met dBASE IV* voor meer informatie over de standaardinstelling van uw printer.

U kunt vanaf de commandostip een ander printerstuurprogramma activeren met de opdracht _pdriver = ''<naam printerstuurprogr>''.

De printer waarnaar het stuurprogramma de uitvoer zendt is de actieve standaardprinter (PRN of LPT1). Mogelijk is dit niet de printer die u in uw bestand Config.db hebt geconfigureerd. Gebruik in dat geval het commando SET PRINTER TO om een andere printer te kiezen.

Gebruik

Printerstuurprogramma's zijn nodig om tekst met een ander schriftsoort af te drukken, bijvoorbeeld **vet**, <u>onderstreept</u> of *cursief*. Alle uitvoer naar de printer verloopt via het opgegeven printerstuurprogramma, behalve de uitvoer van het commando ???.

De naam van het printerstuurprogramma moet voldoen aan de eisen die in DOS worden gesteld aan een bestandsnaam. U kunt ook een padnaam opgeven bij de bestandsnaam. Als u geen extensie opgeeft bij de bestandsnaam, wordt aangenomen dat het bestand de extensie .pr2 heeft.

Printerfonts die zijn geconfigureerd in Config.db, zijn gerelateerd doordat zij het stuurprogramma instellen met _pdriver-opdrachten.

Bijlage F bevat een lijst van de printerstuurprogramma's die worden ondersteund door dBASE. Er kan slechts één printerstuurprogramma tegelijk actief zijn. Als SET TALK op ON is ingesteld, wordt de melding **Printerstuurprogramma geïnstalleerd** weergegeven als het bestand wordt gevonden en de melding **Bestand bestaat niet** als het bestand niet wordt gevonden.

Uitvoer van het commando @ verloopt niet via het printerstuurprogramma.

OPMERKING
Printerstuurprogramma's zijn ontworpen om de tekenset van IBM zo natuurgetrouw mogelijk af te drukken op een bepaalde printer. Wanneer een printer de IBM-tekenset niet ondersteunt, worden bepaalde tekens, bijvoorbeeld lijntekens, vervangen door andere tekens, bijvoorbeeld -, +, = en :. Als bepaalde tekens van de IBM-tekenset niet worden afgedrukt en uw printer deze tekenset wel ondersteunt, controleer dan of het juiste printerstuurprogramma actief is. Met de systeemgeheugenvariabele _pdriver kunt u een ander printerstuurprogramma kiezen.

Opties

Bijlage F bevat een lijst van alle printerstuurprogramma's die kunnen worden gekozen. Wanneer dBASE IV wordt geïnstalleerd, wordt ook een lijst met alle printerstuurprogramma's weergegeven. Kies het bestand Generic.pr2 als uw printer niet in de lijst voorkomt.

U kunt _pdriver instellen op "ASCII" om ASCII-tekstbestanden te maken. In deze bestanden staan geen printerstuurcodes.

Het ASCII-printerstuurprogramma drukt ASCII-tekens af zonder ze voor de printer te vertalen. Wanneer tekens in een dBASE IV-rapport niet goed worden afgedrukt terwijl zij in dBASE III wel goed werden afgedrukt, is het mogelijk dat het printerstuurprogramma deze tekens verkeerd vertaald. Gebruik in zo'n geval het ASCII-printerstuurprogramma voor het afdrukken.

Speciaal geval

Nadat het stuurprogramma voor de PostScript-printer met _pdriver = "POSTSCRI.pr2" is geactiveerd, wordt het bestand PostScript.dld in de printer geladen zodra het eeste teken moet worden afgedrukt. Een paar minuten nadat het afdrukken is beëindigd, is het mogelijk dat de printer opnieuw wordt ingesteld waardoor het dld-bestand opnieuw wordt geladen, geeft u nogmaals het commando _pdriver = "POSTSCRI.pr2".

Voorbeeld

In het volgende programma wordt de gebruiker gevraagd het gewenste printerstuurprogramma te kiezen. Met de functie @M PICTURE wordt een lijst met printerstuurprogramma's weergegeven. Deze lijst kan binnen het venster worden verschoven. Met de opdracht DO CASE wordt _pdriver op de juiste waarde ingesteld nadat de gebruiker een printerstuurprogramma heeft gekozen:

```
Mprinter = SPACE(17)
@ 10,20 SAY "Kies een printer" GET mprinter;
                    PICTURE "@M Epson FX-85,HP LaserJet,IBM QuietWriter,Andere";
                    MESSAGE "Spatiebalk om te bekijken. Return om te kiezen."
READ
DO CASE
    CASE mprinter = "Epson FX-85"
        _pdriver = "FX85_.pr2"
    CASE mprinter = "HP LaserJet"
        _pdriver = "HPLAS100.pr2"
    CASE mprinter = "IBM QuietWriter"
        _pdriver = "IBMQUIET.pr2"
    OTHERWISE
        pdriver = "GENERIC.pr2"
ENDCASE
```

Zie ook

_ppitch, _pquality, ?/??, SET PRINTER

Aan de slag met dBASE IV

_pecode

Met _pecode kunt u printerstuurcodes opgeven die aan het einde van een afdrukopdracht worden verzonden.

Syntaxis

_pecode = <Tuitdr>

Standaardinstelling

Standaard is _pecode een nultekenreeks.

Gebruik

De systeemvariabele _pecode kan alleen worden gebruikt in programma's, omdat de systeemvariabele wordt geactiveerd door het commando ENDPRINTJOB. De variabele moet zijn gedefinieerd voordat het commando ENDPRINTJOB wordt uitgevoerd. Wanneer het commando ENDPRINTJOB wordt uitgevoerd, worden de codes die zijn opgegeven bij _pecode naar de printer gezonden.

Stel dat u een bepaald rapport in een andere schriftsoort dan gewoonlijk wilt afdrukken, bijvoorbeeld een breed rapport in een kleine letter. Met de systeemvariabelen _pscode, die elders in dit hoofdstuk wordt behandeld, en _pecode kunt u bepaalde printerstuurcodes naar de printer zenden om een andere schriftsoort in te stellen. De codes die kunnen worden verzonden, variëren van 0 tot en met 255. De maximale lengte van de codes is 255 tekens.

U kunt ook met de commando's ?/?? en ??? printerstuurcodes naar de printer zenden. Over het algemeen wordt met het commando ?/?? (optie STYLE) de schriftsoort voor delen van de tekst gewijzigd. Met het commando ??? wordt de schriftsoort op een bredere basis in het document gewijzigd. Met de systeemvariabelen _pecode en _pscode wordt de schriftsoort voor het hele document bepaald.

Voorbeeld

U kunt als volgt instellen dat aan het einde van iedere afdrukopdracht de code wordt verzonden waarmee een Hewlett-Packard LaserJet opnieuw wordt ingesteld (raadpleeg het gedeelte over PRINTJOB/ENDPRINTJOB voor meer informatie over afdrukken van gegevens in een programma).

```
_pecode = "{27}{69}"          && Esc E
```

of

```
_pecode = "{27}E"             && Verzendt eveneens Esc E
```

Zie ook

_pscode, ?/?? (STYLE option), ???, PRINTJOB/ENDPRINTJOB

Raadpleeg het boek bij uw printer voor meer informatie over printerstuurcodes.

_peject

Met _peject wordt bepaald of een paginadoorvoerteken (form feed) naar de printer wordt gezonden aan het begin en het einde van een afdrukopdracht.

Syntaxis

_peject = "BEFORE"/"after"/"both"/"none"

Standaardinstelling

De standaardinstelling van _peject is ''BEFORE''.

Gebruik

Het kan zijn dat u een paginadoorvoerteken naar de printer wilt zenden aan het begin of einde van een afdrukopdracht, of zowel aan het begin als aan het einde.

Hoewel u _peject kunt instellen vanaf de commandostip, kunt u deze systeemvariabele alleen gebruiken in programma's. Er moet een afdrukopdracht worden uitgevoerd. Definieer _peject voordat in het programma het commando PRINTJOB wordt uitgevoerd.

Verwar _peject niet met het commando EJECT. Met het commando EJECT kunt u naar het begin van de volgende pagina gaan. Daartoe worden paginadoorvoertekens verzonden of een reeks regeldoorvoertekens, afhankelijk van de instelling van _padvance.

Opties

"BEFORE" — de printkop wordt aan het begin van de volgende pagina geplaatst voordat de eerste pagina wordt afgedrukt.

"AFTER" — nadat de laatste pagina is afgedrukt, wordt de printkop aan het begin van de volgende pagina geplaatst.

"BOTH" — zowel aan het begin als het einde van de afdrukopdracht wordt de printkop aan het begin van de volgende pagina geplaatst.

"NONE" — de printkop wordt noch aan het begin, noch aan het einde van de afdrukopdracht aan het begin van de volgende pagina geplaatst.

Zie ook

_padvance, EJECT, EJECT PAGE, PRINTJOB/ENDPRINTJOB

_pepage

Met _pepage wordt bepaald welke pagina als laatste moet worden afgedrukt.

Syntaxis

_pepage = <Nuitdr>

Standaardinstelling

De standaardinstelling van _pepage is 32767.

Gebruik

Met de systeemvariabele _pepage kunt u instellen dat alle pagina's met een paginanummer dat hoger is dan _pepage niet worden afgedrukt.

De variabele kan een waarde van 1 tot en met 32.767 zijn. De waarde van _pepage kan niet kleiner zijn dan de waarde van _pbpage (zie de bespreking van _pbpage).

Tip

De drie systeemvariabelen _pageno, _pbpage en _pepage moeten met elkaar in overeenstemming zijn. Wanneer bijvoorbeeld _pbpage = 3, _pepage = 5 en _pageno = 10, wordt er geen enkele pagina afgedrukt, omdat het actieve paginanummer buiten het bereik van de af te drukken pagina's ligt (3 tot en met 5).

De waarde van _pageno moet kleiner dan of gelijk zijn aan de waarde van _pepage. In het vorige voorbeeld moet _pageno kleiner dan of gelijk zijn aan 5, anders worden er geen pagina's afgedrukt. Door _pageno, _pbpage en _pepage op de juiste waarden in te stellen, kunt u een bepaald deel van een document afdrukken. Als u bijvoorbeeld één bepaalde pagina wilt afdrukken, stelt u zowel _pbpage als _pepage in op het nummer van de pagina die u wilt afdrukken.

Voorbeeld

U kunt een op uw eigen wensen afgestemd rapport maken zodat er geen records uit het database-bestand meer worden verwerkt zodra _pepage is bereikt. De commandoregel SCAN in het volgende voorbeeld zorgt ervoor dat de verwerking stopt zodra het actieve paginanummer (_pageno) hoger is dan dewaarde van de laatste _pepage. Zonder deze aanpassing met gebruik van de WHILE-clausule zou elk record in het database-bestand Klanten worden gelezen hoewel u had gevraagd slechts de eerste drie pagina's van het rapport af te drukken.

```
USE Klanten
_pageno = 1
_pepage = 3
_plength = 5
PRINTJOB
    SCAN WHILE _pageno<<=_pepage
        ? Achternaam
    ENDSCAN
ENDPRINTJOB
```

Zie ook

_pageno, _pbpage, PRINTJOB/ENDPRINTJOB

_pform

Met _pform kunt u de naam van het actieve afdrukmodel opvragen of een bepaald afdrukmodel activeren.

Syntaxis

_pform = ''<naam afdrukmodel>''

Standaardinstelling

Standaard is _pform een nultekenreeks.

Gebruik

Een afdrukmodel (.prf) is een binair bestand waarin informatie is opgeslagen over de instelling van de printer, in het bijzonder de instelling van de systeemgeheugenvariabelen _padvance, _pageno, _pbpage, _pcopies, _pdriver, _pecode, _peject, _pepage, _plength, _ploffset, _ppitch, _pquality, _pscode, _pspacing en _pwait. Als u een bestemming hebt opgegeven bij het maken van het afdrukmodel, is de DOS-bestandsnaam van de bestemming ook opgeslagen in het bestand, net als de printer zoals die is ingesteld bij PRINTER in Config.db.

U kunt opgeven dat deze instellingen moeten worden opgeslagen in een afdrukmodel wanneer er een rapport- of etikettenbestand wordt opgeslagen door de rapport- of etiketgenerator. De instellingen die zijn opgeslagen in dat afdrukmodel, worden dan automatisch geladen wanneer het bestand wordt bewerkt of afgedrukt vanuit het ontwerpscherm of het Control Center. Vanuit elk menu **Afdrukken** in het Control Center kunt u de instellingen van zo'n afdrukmodel aanpassen. Als u de optie **Opslaan model** kiest, worden de nieuwe instellingen opgeslagen in het .prf-bestand.

Wanneer u echter het commando REPORT FORM of LABEL FORM geeft, wordt het fomulierbestand niet automatisch geladen. Als u bij de commando's REPORT FORM en LABEL FORM toch gebruik wilt maken van de instellingen die zijn opgeslagen in een bepaald afdrukmodel, moet u de naam van dat afdrukmodel toewijzen aan de systeemgeheugenvariabele _pform.

Voorbeeld

U kunt als volgt een bepaald afdrukmodel activeren voordat u een rapport afdrukt met het commando REPORT FORM:

```
. _pform = "RapportA"
. REPORT FORM Rapport
```

Zie ook

CREATE/MODIFY LABEL, CREATE/MODIFY REPORT, LABEL FORM, REPORT FORM

_plength

Met _plength wordt de lengte van het printerpapier bepaald.

Syntaxis

_plength = <Nuitdr>

Standaardinstelling

De standaardlengte van een pagina is 66 regels.

Gebruik

De totale lengte van een pagina is het aantal regels dat kan worden afgedrukt tussen de bovenkant en de onderkant van de pagina.

De systeemvariabele _plength kan lopen van 1 tot en met 32.767.

U hoeft de bovenmarge en ondermarge van de pagina niet op te geven. Met _plength bepaalt u de totale lengte van de pagina. Met het commando ON PAGE kunt u instellen op welke positie de kopteksten, voetteksten en normale tekst worden afgedrukt.

OPMERKING
In afbeelding 1-1 in hoofdstuk 1, "Basisbegrippen", wordt de instelling van _plength weergegeven.

Voorbeeld

In het voorbeeld dat wordt gegeven bij de systeemvariabele _padvance in dit hoofdstuk, wordt met de systeemvariabele _plength de lengte van de pagina ingesteld.

Zie ook

_padvance, EJECT, EJECT PAGE, ON PAGE

_plineno

Met _plineno kunt u het regelnummer voor de uitvoerstroom instellen of het actieve regelnummer opvragen.

Syntaxis

_plineno = <Nuitdr>

Standaardinstelling

De standaardinstelling van _plineno is 0.

Gebruik

De instelling van _plineno heeft betrekking op alle gegevens die worden uitgevoerd. Deze *uitvoerstroom* wordt geproduceerd door alle commando's die gegevens uitvoeren, met uitzondering van de commando's @, @...TO en EJECT. De bestemming van de uitvoerstroom kan worden bepaald met de commando's SET CONSOLE, SET PRINTER en SET ALTERNATE en met de opties TO PRINTER/TO FILE <bestandsnaam> van commando's als LIST/DISPLAY. De waarde van _plineno wordt verhoogd, ongeacht of de uitvoerstroom naar de printer of naar het scherm wordt gestuurd. Het invoeren van gegevens op het scherm en het gebruik van ↵ beïnvloeden daarom ook het actieve regelnummer.

Het actieve regelnummer van uw document wordt automatisch bijgehouden. Deze variabele wordt verhoogd terwijl de tekst wordt afgedrukt en ingedeeld.

Als de positie van het printerpapier is gewijzigd, kunt u de variabele _plineno instellen op het nummer van de regel waarop de printkop staat. De variabele _plineno kan variëren van 0 tot en met (_plength - 1).

In tegenstelling tot PROW() blijft _plineno ook van kracht als u de printer uitschakelt en de gegevens op het scherm verschuift. Een ander verschil is dat de waarde van _plineno nooit groter kan zijn dan de waarde van _plength.

Verwar de systeemgeheugenvariabele _plineno niet met de functie LINENO(). Met de functie LINENO() kunt u opvragen welke regel van een programma zal worden uitgevoerd.

Zie ook

_plength, EJECT PAGE, ON PAGE, PCOL(), PROW()

_ploffset

Met _ploffset wordt de linker marge van de printer bepaald.

Syntaxis

_ploffset = <Nuitdr>

Standaardinstelling

De linker marge van de printer is standaard ingesteld op 0.

Gebruik

De linker marge van de printer wordt gemeten vanaf de linker zijde van het papier. _lmargin begint waar _ploffset eindigt. Met deze systeemvariabele wordt bepaald hoeveel spaties links van iedere regel worden afgedrukt. Met de systeemvariabele _ploffset kunt u alle tekst op een pagina naar rechts verschuiven. Met deze functie kunt u eenvoudig de positie van de afgedrukte tekst wijzigen wanneer het papier niet precies in het midden van de printerwagen staat.

Deze variabele kan variëren van 0 tot en met 254.

Met het commando SET MARGIN wordt de instelling van de systeemvariabele _ploffset gewijzigd, niet de instelling van de systeemvariabele _lmargin.

De instellingen van _ploffset en _lmargin zijn onafhankelijk van elkaar. Stel dat voor een rapport _ploffset op 10 wordt ingesteld en _lmargin op 5. Voor een deel van het rapport wordt _indent ingesteld op 10. Wanneer nu de waarde van _ploffset wordt gewijzigd in 5, wordt de verhouding tussen _lmargin en _indent niet gewijzigd.

OPMERKING
In afbeelding 1-1 in hoofdstuk 1, "Basisbegrippen", wordt de instelling _ploffset weergegeven.

Zie ook

_lmargin, SET MARGIN

_ppitch

Met _ppitch kunt u de tekenbreedte van de printer instellen of een tekenreeks opvragen met de op dat moment ingesteld tekenbreedte.

Syntaxis

_ppitch = "pica"/"elite"/"condensed"

Standaardinstelling

De standaardinstelling van _ppitch is DEFAULT. DEFAULT is de standaardtekenbreedte die bij het starten van dBASE IV wordt bepaald door de DIP-schakelaars of opstartcodes van uw printer.

Wanneer u _ppitch hebt gewijzigd in "PICA", "ELITE" of "CONDENSED" kunt u de waarde niet opnieuw instellen op "DEFAULT". Het volgende voorbeeld illustreert dit.

```
. ? _ppitch
DEFAULT
. _ppitch = "ELITE"
. ? _ppitch
ELITE
. _ppitch = "DEFAULT"
. ? _ppitch
ELITE
```

Aangezien de instelling van _ppitch "DEFAULT" is, worden er geen codes naar de printer gestuurd en blijft de oude instelling "ELITE" actief.

Gebruik

Met _ppitch kunt u de tekenbreedte van uw printer (tekens per inch) instellen door de juiste stuurcode naar het actieve printerstuurprogramma te zenden (kies het printerstuurprogramma met de systeemvariabele _pdriver).

"PICA" — 10 tekens per inch.

"ELITE" — 12 tekens per inch.

"CONDENSED" — ongeveer 17,16 tekens per inch.

"DEFAULT" — ongewijzigde printerinstelling.

De ingestelde optie moet worden ondersteund door het actieve printerstuurprogramma.

Voorbeelden

U kunt op de volgende manier opvragen welke tekenbreedte is ingesteld:

```
. ? _ppitch
```

In het volgende deel van een programma wordt de tekenbreedte voor de kop- en voetteksten ingesteld op ELITE. De tekenbreedte wordt voor de koppen van de kolommen ingesteld op PICA en voor de normale tekst van het rapport op CONDENSED. Als het rapport is afgedrukt, wordt de tekenbreedte opnieuw ingesteld op PICA. De procedures Koptekst, Kol_kop, P_tekst en Voettkst staan niet op de diskettes van dBASE IV.

```
_ppitch = "ELITE"

DO WHILE .NOT. EOF()
    DO Koptekst
        _ppitch = "PICA"
    DO Kol_kop
        _ppitch = "CONDENSED"
    DO P_tekst
        _ppitch = "ELITE"
    DO Voettkst
ENDDO
_ppitch = "PICA"
```

Zie ook

_pdriver, _pquality

_pquality

Met _pquality kunt u de printer instellen op de conceptmodus (draft) of de correspondentiemodus (NLQ). U kunt met deze systeemvariabele ook een logische waarde opvragen die aangeeft op welke modus de printer is ingesteld.

Syntaxis

_pquality = <voorwaarde>

Default

Standaard is de systeemvariabele _pquality ingesteld op onwaar (.F.), dat wil zeggen conceptmodus.

Gebruik

Als _pquality op waar (.T.) is ingesteld, wordt de correspondentiemodus gekozen, letter-quality of near letter-quality, afhankelijk van de printer. Als _pquality op onwaar (.F.) is ingesteld, wordt de conceptmodus gekozen (met _pdriver kunt u een printerstuurprogramma kiezen, zodat de juiste stuurcodes naar de printer worden gezonden).

In de correspondentiemodus is de kwaliteit (resolutie) van de afdruk hoger. In de conceptmodus wordt een document veel sneller afgedrukt. Overigens ondersteunen niet alle printers zowel conceptmodus als correspondentiemodus.

Tip

Als de kwaliteitsinstelling op de printer zelf wordt gewijzigd, kan het zijn dat de kwaliteitsinstelling van de printer niet meer overeenkomt met de instelling van _pquality. In dat geval heeft de instelling van de printer een hogere prioriteit. Geef de opdracht _pquality vanaf de commandostip of vanuit een programma als u er zeker van wilt zijn dat de twee instellingen overeenkomen.

Zie ook

_pdriver, _ppitch

In bijlage F, “dBASE IV-printerstuurprogramma’s” vindt u een overzicht van de beschikbare printerstuurprogramma’s en de ondersteunde modi.

_pscode

Met _pscode kunt u opgeven welke stuurcodes moeten worden verzonden aan het begin van een afdrukopdracht.

Syntaxis

_pscode = <Tuitdr>

Standaardinstelling

Standaard is _pscode een nultekenreeks.

Gebruik

Het kan voorkomen dat u een bepaald rapport in een andere schriftsoort wilt afdrukken dan gewoonlijk, bijvoorbeeld een breed rapport in een kleine letter. Met de systeemvariabelen _pscode kunt u bepaalde printerstuurcodes naar de printer zenden om een andere schriftsoort in te stellen aan het begin van de printopdracht (met de systeemvariabele _pecode die elders in dit hoofdstuk wordt behandeld, kunt u de standaard schriftsoort weer instellen).

Raadpleeg het gedeelte over het commando ??? en het boek bij uw printer voor meer informatie over printerstuurcodes.

U kunt _pscode instellen vanaf de commandostip of vanuit een programma. In een programma moet _pscode worden ingesteld voor het commando PRINTJOB wordt uitgevoerd. Wanneer het commando PRINTJOB wordt uitgevoerd, worden de codes die zijn opgegeven bij _pscode naar de printer gezonden. U moet alle codes tussen accolades plaatsen. De lengte van de <Tuitdr> kan variëren van 0 tot en met 255 tekens.

U kunt ook met de commando's ?/?? en ??? printerstuurcodes naar de printer zenden. Over het algemeen wordt met het commando ?/?? (de optie STYLE) de schriftsoort van afzonderlijke regels tekst gewijzigd. Met het commando ??? wordt in een document de schriftsoort op een bredere basis gewijzigd. Met de systeemvariabelen _pscode en _pecode kunt u een schriftsoort kiezen voor het hele document.

Voorbeeld

In het volgende voorbeeld wordt een Hewlett-Packard LaserJet aan het begin van een afdrukopdracht ingesteld op een liggende oriëntatie (raadpleeg het gedeelte over de commando's PRINTJOB/ENDPRINTJOB voor meer informatie over het afdrukken van gegevens in programma's).

```
_pscode = {27}{38}{108}{49}{79}
```

Zie ook

_pecode, ?/?? (STYLE option), ???, PRINTJOB/ENDPRINTJOB

_pspacing

Met _pspacing kunt u de regelafstand van de uitvoer instellen.

Syntaxis

_pspacing = 1/2/3

Standaardinstelling

De standaardregelafstand is 1.

Gebruik

Met de systeemvariabele _pspacing kunt u de regelafstand instellen voor de uitvoerstroom. De regelafstand is het aantal regels tussen de regels van de uitvoer. _pspacing kan worden ingesteld op 1, 2 of 3.

De instelling van _pspacing is ook van belang voor de hoogte van een kader. Als _pspacing is ingesteld op 2 en voor een kader de hoogte 10 is opgegeven, wordt het kader afgedrukt met een hoogte van 20.

Wanneer u bij wijze van uitzondering een andere regelafstand wenst, gebruikt u het commando ?.

Opties

1 — Enkele regelafstand: geen witregels.

2 — Dubbele regelafstand: één witregel.

3 — Driedubbele regelafstand: twee witregels.

Voorbeeld

In het volgende programma wordt dubbele regelafstand ingesteld voor de uitvoer van het commando LIST:

```
_pspacing = 2
LIST TO PRINT
```

Zie ook

?/??, LIST/DISPLAY

_pwait

Met _pwait kunt u opgeven of de printer al dan niet stopt nadat een pagina is afgedrukt.

Syntaxis

_pwait = <voorwaarde>

Standaardinstelling

De standaardinstelling van _pwait is onwaar (.F.).

Gebruik

Met behulp van de systeemvariabele _pwait kunt u op losse vellen papier afdrukken. Als _pwait op waar (.T.) is ingesteld, stopt de printer nadat een pagina is afgedrukt en kunt u een nieuw vel papier invoeren.

De printer stopt als een pagina is doorgevoerd met het commando EJECT, of als er meer regels zijn afgedrukt dan het maximale aantal regels dat is opgegeven bij _plength.

Wanneer _pwait wordt gebruikt in een afdrukopdracht, moet de systeemvariabele zijn ingesteld voordat het commando PRINTJOB wordt uitgevoerd.

Voorbeeld

Als u _pwait instelt op waar (.T.), pauzeert de printer tussen de pagina's:

```
. _pwait = .T.
```

Zie ook

_plength, EJECT, EJECT PAGE

_rmargin

Met _rmargin wordt de rechter marge ingesteld voor de uitvoer van het commando ?/??. _rmargin is alleen actief als _wrap op waar (.T.) is ingesteld.

Syntaxis

_rmargin = <Nuitdr>

Standaardinstelling

De rechter marge is standaard ingesteld op 79 tekens.

Gebruik

De rechter marge is de eerste kolom voorbij de tekst die op een bepaalde regel wordt afgedrukt.

De minimumwaarde van _rmargin is (_lmargin+1) of (_lmargin+_indent+1), afhankelijk van welke van de twee het grootste is. _indent mag natuurlijk een negatief getal zijn. De maximumwaarde van _rmargin is 255.

Stel dat zowel _lmargin als _indent zijn ingesteld op 5. De minimumwaarde van _rmargin is dan 10, zodat ten minste één kolom kan worden afgedrukt.

Als _wrap is ingesteld op onwaar (.F.), heeft de instelling van _rmargin geen invloed.

OPMERKING
In afbeelding 1-1 in hoofdstuk 1, "Basisbegrippen", wordt de instelling van _rmargin weergegeven.

Voorbeeld

Met _rmargin kunt u vanaf de commandostip of vanuit een programma de breedte van een alinea verkleinen. In het volgende voorbeeld wordt dezelfde tekst twee keer afgedrukt met verschillend ingestelde marges:

```
SET TALK OFF

_wrap = .T.

_rmargin = 72
_lmargin = 0

Mtekst = "Mijn hoed die heeft drie deuken. Drie deuken " +;
"heeft mijn hoed. En had hij niet drie deuken, dan was " +;
"het niet mijn hoed."

? Mtekst
?

_rmargin = 45
_lmargin = 15

? Mtekst
```

De uitvoer van deze routine ziet er als volgt uit:

```
Mijn hoed die heeft drie deuken. Drie deuken heeft mijn hoed. En had hij niet drie deuken, dan was het niet mijn hoed.

                    Mijn hoed die heeft drie
                    deuken. Drie deuken heeft mijn
                    hoed. En had hij niet drie
                    deuken, dan was het niet mijn
                    hoed.
```

Zie ook

_indent, _lmargin, _ploffset, _wrap

_tabs

Met _tabs kunt u één of meer tabs instellen voor het scherm, de printer of de uitvoer van een bestand dat wordt afgedrukt met het commando ?/??. Met deze systeemvariabele worden ook de tabstops voor de tekstverwerker ingesteld.

Syntaxis

_tabs = <Tuitdr>

Standaardinstelling

Hoewel _tabs standaard op een lege tekenreeks is ingesteld, zijn er toch tabs standaard ingesteld op 8, 16, 24, 32, enzovoort.

Gebruik

Met _tabs kunt u een lijst met tabstops opgeven. Wanneer een tabteken, CHR(9), wordt afgedrukt met het commando ?/??, worden net zoveel spaties afgedrukt als nodig zijn om de volgende tabstop te bereiken.

De lijst met tabstops bestaat uit een serie cijfers in oplopende volgorde, gescheiden door komma's. Elk cijfer vertegenwoordigt het kolomnummer van een tabstop.

U kunt altijd tabs instellen, ongeacht of _wrap nu op waar (.T.) of op onwaar (.F.) is ingesteld. Als _wrap op waar (.T.) is ingesteld, worden tabstops waarvan het kolomnummer hoger dan of gelijk is aan _rmargin genegeerd.

Met _tabs kunt u ook tabstops instellen voor de tekstverwerker. Als er geen tabs zijn ingesteld met _tabs, zijn de tabs in de tekstverwerker ingesteld op acht spaties per tabstop.

_tabs is gelijk aan de instelling van TABS in het bestand Config.db.

Voorbeeld

Met het volgende commando worden de tabstops ingesteld op 5, 20, 36 en 60:

```
. _tabs = "5, 20, 36, 60"
```

Zie ook

_indent, _wrap

Het bestand Config.db wordt behandeld in *Aan de slag met dBASE IV*.

_wrap

Met _wrap kunt u de automatische regelovergang in- en uitschakelen.

Syntaxis

_wrap = <voorwaarde>

Standaardinstelling

Standaard is _wrap ingesteld op onwaar (.F.).

Gebruik

Wanneer _wrap wordt ingesteld op waar (.T.), wordt bij de uitvoer van de commando's ? en ?? rekening gehouden met de marges die zijn ingesteld met _lmargin en _rmargin. Tekst die voorbij de rechter marge doorloopt, wordt nu automatisch op de volgende regel afgedrukt of weergegeven. De regels worden afgebroken tussen woorden of getallen.

De systeemgeheugenvariabelen _alignment, _indent, _lmargin en _rmargin hebben alleen invloed als _wrap op waar (.T.) is ingesteld.

Wanneer _wrap op waar (.T.) is ingesteld, wordt de uitvoerstroom in een buffer opgeslagen totdat de actieve regel is afgewerkt. Als u de uitvoer start met een commando ?, moet u mogelijk nogmaals een commando ? geven zodat ook de laatste regel van de tekst wordt afgedrukt. In het volgende voorbeeld is dit beschreven:

Voorbeeld

Met de systeemvariabele _wrap wordt bepaald of de marges en de uitlijn- en inspringfunctie worden geactiveerd:

```
_indent = 3
_lmargin = 10
_rmargin = 30
Mem = "Dit is een zin die uit bijna vijftig tekens bestaat."
_wrap = .T.
? Mem
?
```

De uitvoer van het voorbeeld ziet er als volgt uit:

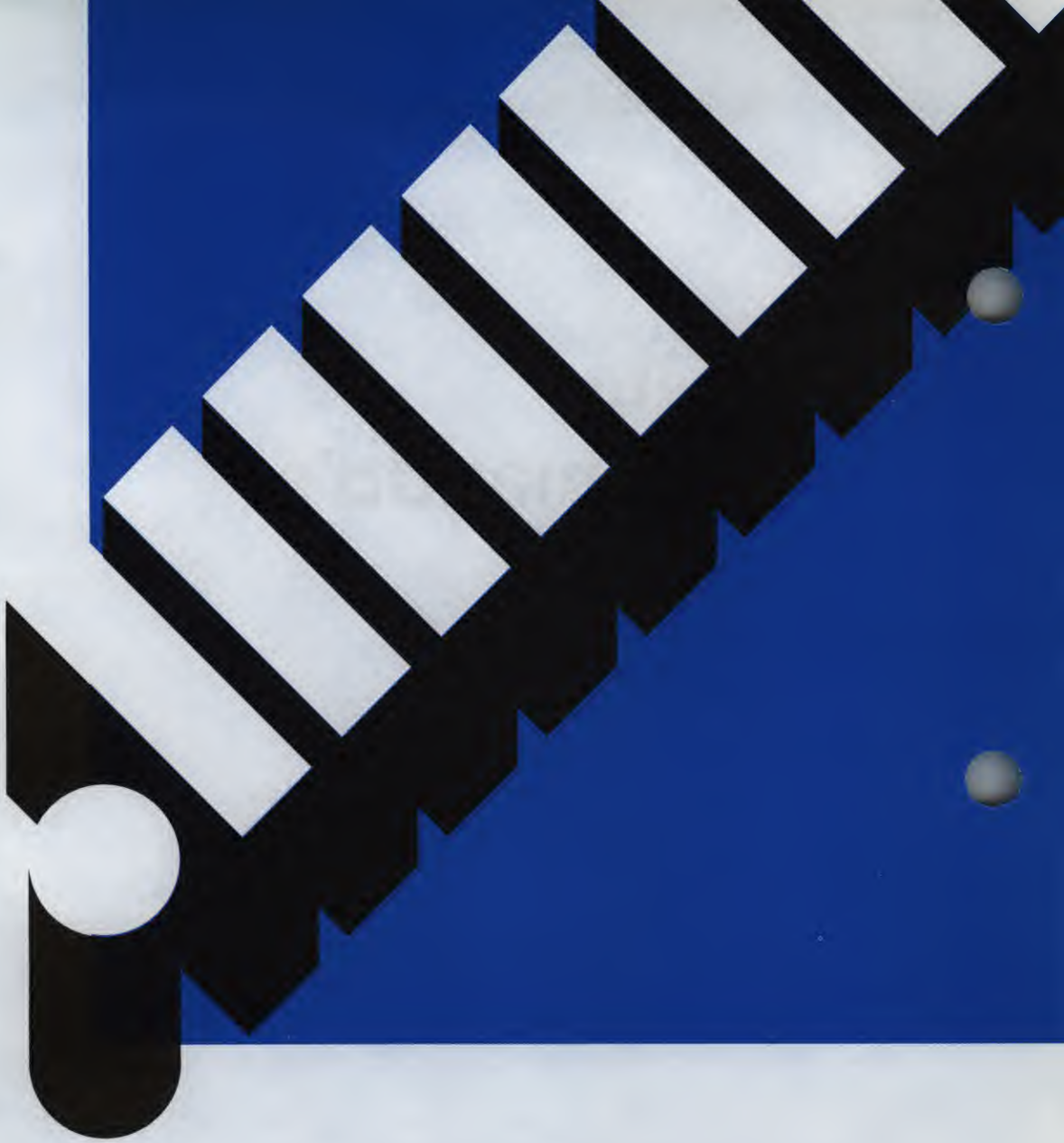
```
          Dit is een zin die
uit bijna vijftig
tekens bestaat.
```

Zie ook

_alignment, _indent, _lmargin, _rmargin

Naslagshandboek van dBASE IV

SQL-commando's



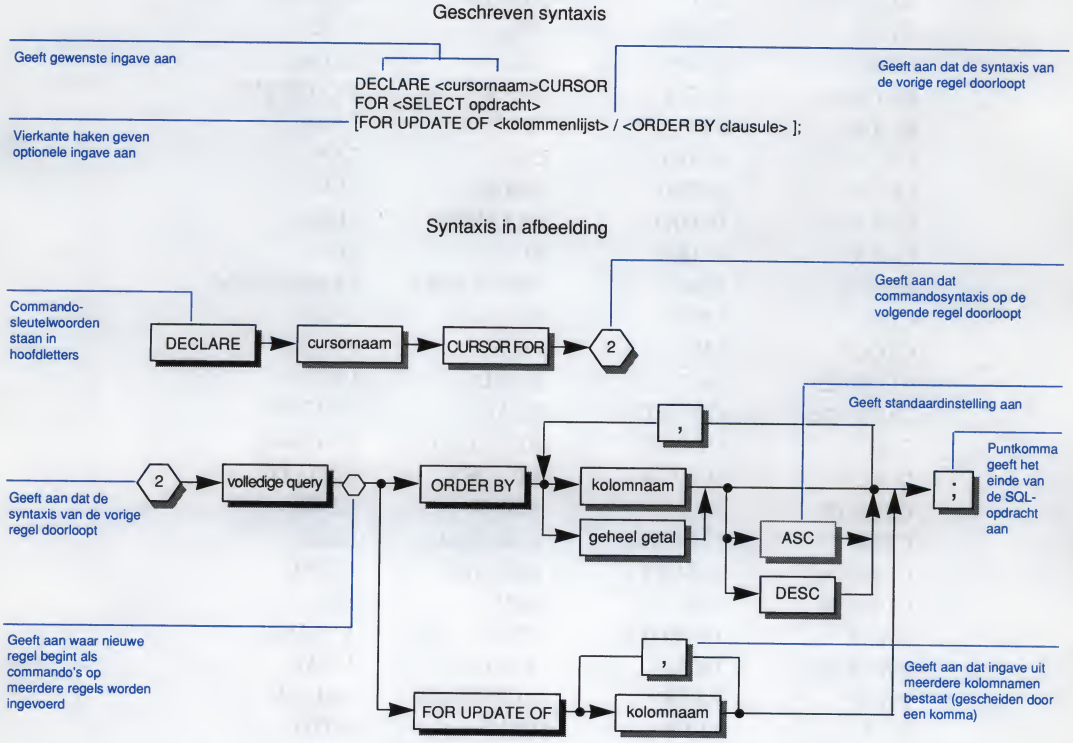
i 1 2 3 4 5 6 7 A B
C D E F G H In

SQL-commando's

In dit hoofdstuk worden de syntaxis en het gebruik van de SQL-commando's waarover u met dBASE IV beschikt, behandeld. De commando's zijn alfabetisch gerangschikt.

Symbolen en conventies

De syntaxis van de SQL-commando's wordt in geschreven vorm en in de vorm van afbeeldingen behandeld. In afbeelding 6-1 wordt uitgelegd hoe u de beide varianten van de syntaxis moet interpreteren.



Afbeelding 6-1 Syntaxis van de commando's

Gereserveerde woorden

In tabel 6-1 staan alle woorden die alleen door SQL kunnen worden gebruikt. Deze woorden hebben een bijzondere betekenis en mogen niet als naam voor een tabel, visie, index, kolom, geheugenvariabele of als synoniem worden gebruikt.

Tabel 6-1 Gereserveerde SQL-woorden

ABS	DECIMAL	LOG	SHOW
ACOS	DECLARE	LOG10	SIGN
ADD	DELETE	LOGICAL	SIN
ALL	DELIMITED	LOWER	SMALLINT
ALTER	DESC	LTRIM	SOUNDEX
AND	DIF	MAX	SPACE
ANY	DIFFERENCE	MDY	SQRT
AS	DISTINCT	MIN	START
ASC	DMY	MOD	STOP
ASIN	DOW	MONTH	STR
AT	DROP	NOT	STUFF
ATAN	DTOC	NUMERIC	SUBSTR
ATN2	DTOR	OF	SUM
AVG	DTOS	ON	SYLK
BETWEEN	EXISTS	OPEN	SYNONYM
BLANK	EXP	OPTION	TABLE
BY	FETCH	OR	TAN
CDOW	FIXED	ORDER	TEMP
CEILING	FLOAT	PAYMENT	TIME
CHAR	FLOOR	PI	TO
CHECK	FOR	PRIVILEGES	TRANSFORM
CHR	FROM	PUBLIC	TRIM
CLOSE	FV	PV	TYPE
CLUSTER	FW2	RAND	UNION
CMONTH	GRANT	REAL	UNIQUE
COS	GROUP	REPLICATE	UNLOAD
COUNT	HAVING	REVOKE	UPDATE
CREATE	IN	RIGHT	UPPER
CTOD	INDEX	ROLLBACK	USER
CURRENT	INSERT	ROUND	USING
CURSOR	INT	RPD	VAL
DATA	INTEGER	RTOD	VALUES
DATBASE	INTO	RTRIM	VIEW
DATE	KEEP	RUNSTATS	WHERE
DAY	LEFT	SAVE	WITH
DBASEII	LEN	SDF	WKS
DBCHECK	LIKE	SELECT	WORK
DBDEFINE	LOAD	SET	YEAR

OPMERKING
U moet de namen van dBASE IV-commando's en -functies in SQL niet als namen voor tabellen, visies, indexen, kolommen, geheugenvariabelen of als synoniemen gebruiken.

Commandocategorieën

De SQL-commando's in dBASE IV kunnen worden ingedeeld naar het soort bewerking dat ermee wordt uitgevoerd. De SQL-commando's worden in dit gedeelte per categorie vermeld. Zoals u ziet, vallen sommige commando's in meer categorieën.

SQL-databases maken en opstarten

CREATE DATABASE	Maakt een database-bestandsindex en een set SQL-catalogustabellen voor de nieuwe database
SHOW DATABASE	Toont een lijst met de beschikbare databases
START DATABASE	Opent een database (maakt deze actief)
STOP DATABASE	Sluit de actieve database

Objecten maken en wijzigen

ALTER TABLE	Voegt één of meer nieuwe kolommen toe aan een bestaande tabel
CREATE INDEX	Maakt een index op basis van de waarden van één of meer kolommen in een tabel of visie
CREATE SYNONYM	Definieert een alternatieve naam voor een tabel of visie
CREATE TABLE	Maakt een nieuwe tabel en definieert de kolommen van die tabel
CREATE VIEW	Maakt een virtuele tabel op basis van de kolommen in andere tabellen of visies

Beveiliging van databases

GRANT	Verleent een gebruiker toegangs- en bijwerkrechten voor een tabel
REVOKE	Trekt de met GRANT verleende rechten weer in

Gegevens definiëren

ALTER TABEL	Voegt één of meer nieuwe kolommen aan een bestaande tabel toe
CREATE DATABASE	Maakt een database-bestandsindex en een set SQL-catalogustabellen voor de nieuwe database
CREATE INDEX	Maakt een index op basis van de waarden van één of meer kolommen in een tabel of visie
CREATE SYNONYM	Definieert een alternatieve naam voor een tabel of visie
CREATE TABLE	Maakt een nieuwe tabel en definieert de kolommen van die tabel
CREATE VIEW	Maakt een virtuele tabel op basis van de kolommen in andere tabellen of visies
DBDEFINE	Maakt ingaven voor een SQL-catalogustabel bij database-bestanden die als SQL-tabellen zijn gedefinieerd
DROP DATABASE	Verwijdert een SQL-database met alle bijbehorende objecten uit de database-bestandsindex
DROP INDEX	Verwijdert een index
DROP SYNONYM	Verwijdert een synoniem
DROP TABLE	Verwijdert een tabel
DROP VIEW	Verwijdert een visie
RUNSTATS	Werkt de statistieken van catalogustabellen bij om de toegankelijkheid van de SQL-tabellen te optimaliseren

Objecten verwijderen

DROP DATABASE	Verwijdert een SQL-database met alle bijbehorende objecten uit de database-bestandsindex
DROP INDEX	Verwijdert een index
DROP SYNONYM	Verwijdert een synoniem
DROP TABLE	Verwijdert een tabel
DROP VIEW	Verwijdert een visie

Ingebedde SQL

CLOSE	Maakt een SQL-cursor inactief
DECLARE CURSOR	Definieert een cursor en initieert een SELECT-commando waarmee de voor de cursor beschikbare rijen worden gedefinieerd
FETCH	Verplaatst de cursorwijzer naar de volgende met SELECT geselecteerde rij en kopieert de waarden uit die rij naar geheugenvariabelen van dBASE
OPEN	Opent een cursor en voert het SELECT-commando bij de bijbehorende cursor uit

Gegevens bijwerken en een query uitvoeren

DELETE	Verwijdert opgegeven rijen uit een tabel of een bewerkbare visie
INSERT	Voegt nieuwe rijen toe aan een tabel of een bewerkbare visie
SELECT	Haalt gegevens op uit rijen van één of meer tabellen of visies
UPDATE	Wijzigt de gegevens in opgegeven rijen van een tabel of een bewerkbare visie

Hulpprogramma's

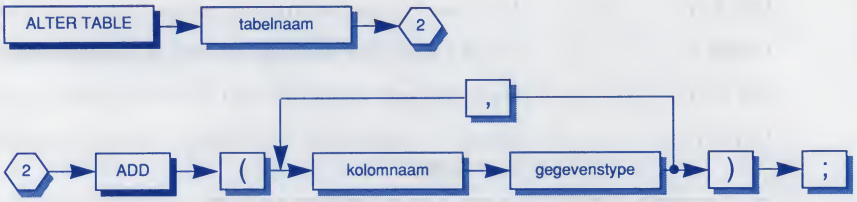
DBCHECK	Controleert elementen uit een SQL-catalogustabel bij database-tabellen
DBDEFINE	Maakt ingaven voor een SQL-catalogustabel bij database-bestanden die als SQL-tabellen zijn gedefinieerd
LOAD DATA	Importeert gegevens uit een bestand in een SQL-tabel
RUNSTATS	Werkt de statistieken van catalogustabellen bij om de toegankelijkheid van de SQL-tabellen te optimaliseren
UNLOAD DATA	Exporteert gegevens uit een SQL-tabel naar een ander bestandsformaat

ALTER TABLE

Met het commando ALTER TABLE kunt u één of meer kolommen aan een tabel in de actieve database toevoegen. Als dBASE IV met een wachtwoord is beveiligd, moet u zelf de te wijzigen tabel hebben gemaakt of over het ALTER-recht voor die tabel beschikken

Syntaxis

ALTER TABLE <tabelnaam>
    ADD (<kolomnaam><gegevenstype>
    [,<kolomnaam><gegevenstype>...]);



Afbeelding 6-2 Het commando ALTER TABLE

Gebruik

Wanneer u een kolom aan een tabel toevoegt, moet u de naam en het gegevenstype van die kolom opgeven. Bij alle kolommen van het type CHAR en bij sommige van het type NUMERIC moet u ook de breedte van de kolom en het aantal decimale posities opgeven.

In tabel 6-2 vindt u een overzicht van de gegevenstypen die u bij het commando ALTER TABLE kunt gebruiken om kolommen te definiëren. Het zijn dezelfde gegevenstypen waarmee u bij CREATE TABLE kolommen definieert. Bij beide commando's worden de gegevenstypen volgens dezelfde regels gebruikt. Raadpleeg het gedeelte over CREATE TABLE voor een nauwkeuriger beschrijving van de gegevenstypen van SQL.

Tabel 6-2 Gegevenstypen van SQL

Gegevenstype	Omschrijving
CHAR (n)	Gegevens bestaand uit tekenreeksen
DATE	Datums (standaardindeling: mm/dd/jj)
DECIMAL (x,y)	Getallen met een vast decimaalteken
FLOAT (x,y)	Getallen met zwevende komma
INTEGER	Geheel getal van 11 cijfers
LOGICAL	Logische waarde (waar of onwaar)
NUMERIC (x,y)	Getallen met een vast decimaalteken
SMALLINT	Geheel getal van 6 cijfers

OPMERKING

1. *Wanneer u nieuwe kolommen toevoegt, moet u er rekening mee houden dat het totale aantal kolommen niet groter mag zijn dan 255 en dat de totale breedte van alle velden niet meer dan 4000 bytes in beslag mag nemen.*
2. *U kunt met dBASE IV-SQL geen dBASE IV-memovelden gebruiken of maken.*
3. *Nieuwe kolommen van het type CHAR die aan bestaande rijen worden toegevoegd, worden geinitialiseerd met blanco posities. Nieuwe kolommen van het type NUMERIC worden op 0 gesteld. Nieuwe kolommen van het type LOGICAL worden op onwaar (.F.) ingesteld.*

Elke nieuwe kolom wordt rechts van de laatst gedefinieerde kolom van een tabel toegevoegd. U kunt kolommen alleen aan een tabel en niet aan een visie toevoegen. Evenmin kunt u een bestaande kolom verwijderen of de definitie ervan wijzigen.

TIP

Om een kolom uit een tabel te verwijderen of het gegevenstype van een kolom te wijzigen moet u een nieuwe tabel met nieuwe kolomdefinities maken. Vervolgens kunt u de gegevens uit de betreffende kolom uit de oude tabel aan de nieuwe toevoegen met behulp van de vorm <SELECT-commando> van het commando INSERT.

ALTER TABLE

Voorbeeld

Wanneer u de nieuwe kolommen Tel_nr en Ltste_best aan de tabel Klant wilt toevoegen typt u:

```
SQL.ALTER TABLE Klant
      ADD (Tel_nr CHAR(13), Ltste_best DATE);
```

Zie ook

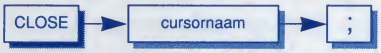
CREATE TABLE, DROP TABLE, INSERT

CLOSE

Met het commando CLOSE kunt u in de ingebedde SQL-modus een geopende cursor inactief maken.

Syntaxis

CLOSE <cursornaam>;



Afbeelding 6-3 Het commando CLOSE

Gebruik

Het commando DECLARE CURSOR wordt in een ingebed SQL-programma gebruikt om de cursor te definiëren en het commando SELECT, waarmee de resultaattabel van de cursor wordt gemaakt, te initiëren. Vervolgens wordt met het commando OPEN de cursor actief gemaakt, waarna het commando SELECT wordt uitgevoerd om de resultaattabel te produceren. Door het commando CLOSE te geven maakt u de cursor inactief en wordt de geheugenruimte vrijgemaakt die door de resultaattabel in beslag wordt genomen.

De cursor die u opgeeft bij <cursornaam> moet al geopend zijn. U kunt een inactief gemaakte cursor opnieuw openen door nogmaals het commando OPEN te geven. Als u een cursor opnieuw opent, wordt het commando SELECT van die cursor nogmaals uitgevoerd voor een nieuwe resultaattabel.

In de volgende gevallen wordt een cursor automatisch inactief gemaakt:

- Het programma draagt de besturing over aan de commandostip of aan de SQL-aanwijzing.
- Het commando SET SQL OFF wordt uitgevoerd.
- Het actieve databestand wordt gesloten met CREATE DATABASE, START DATABASE of STOP DATABASE.

Een cursor blijft echter geopend wanneer een ingebedde SQL-programmamodule een andere module aanroept. Wanneer de uitvoering van de aanroepende module wordt voortgezet, verwijst de cursor naar de oorspronkelijke rij.

Om compileerfouten te vermijden moet ieder commando dat naar de cursor verwijst, worden gevolgd door het commando CLOSE .

TIP
U kunt een cursor zo vaak als u wenst openen en weer inactief maken om het SELECT-commando van de cursor uit te voeren. Een cursor moet inactief worden gemaakt voordat deze opnieuw kan worden geopend.

CLOSE

Voorbeeld

In het volgende ingebedde SQL-programma wordt met DECLARE de cursor Fact gemaakt, waarna het bijbehorende commando SELECT wordt gedefinieerd. Door het commando OPEN wordt Fact geopend en SELECT uitgevoerd.

Nadat met FETCH gegevens uit de resultaattabel zijn opgehaald, gebruikt het programma Factuur deze om facturen te maken. Vervolgens worden uit de tabel Verkoop rijen verwijderd waarin de orders zijn gefactureerd. Ten slotte wordt met het commando CLOSE de cursor Fact gesloten, waarna de geheugenruimte die wordt gebruikt voor de resultaattabel van SELECT, weer wordt vrijgemaakt.

```
DECLARE Fact CURSOR FOR
 SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
 FROM Verkoop
 WHERE NOT Gefact;
   .
   .
   .
OPEN Fact;
   .
   .
   .
 DO WHILE .T.
  FETCH Fact INTO morder_nr, mverk_dat, mmedew_nr, mklant_nr, mgefact;
  IF SQLCODE = 0
   * Druk een factuur voor ordernummer in de huidige rij af
   * Indien gelukt, wijzig geheugenvariabele mgefact in .T.
   *
   DO Factuur WITH morder_nr, mverk_dat, mmedew_nr, mklant_nr, ; mgefact
   *
   IF mgefact .AND. Verk_dat < CTOD("22/09/90")
    DELETE FROM Verkoop
    WHERE CURRENT OF Fact;
    ENDIF
   ELSE
   EXIT
  ENDIF
 ENDDO
CLOSE Fact;
```

Zie ook

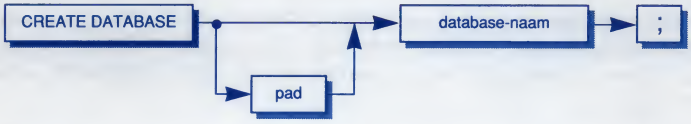
DECLARE CURSOR, FETCH, OPEN

CREATE DATABASE

Met het commando CREATE DATABASE wordt een nieuwe database gemaakt, waarin nieuwe database-objecten zoals tabellen, visies, synoniemen en indexen kunnen worden ondergebracht. Om een database te maken hebt u geen speciale rechten nodig.

Syntaxis

CREATE DATABASE [<pad>]<database-naam>;



Afbeelding 6-4 Het commando CREATE DATABASE

Gebruik

Met het commando CREATE DATABASE kunt u een SQL-database maken en een set SQL-catalogustabellen in de bestandsindex van de nieuwe database definiëren. Alle gegevens over objecten die bij de database worden gemaakt, worden in de catalogustabellen opgeslagen.

OPMERKING
Voordat u het commando CREATE DATABASE gebruikt, moet u ervoor zorgen dat de opdracht SQLHOME = <padnaam> in het bestand Config.db is opgenomen. In deze opdracht wordt de bestandsindex waarin de SQL-bestanden staan, vastgelegd. Raadpleeg het boek Aan de slag met dBASE IV.

<database-naam> kan een willekeurige in dBASE IV geldige naam zijn. Dat wil zeggen dat de naam:

- uit maximaal acht tekens mag bestaan;
- uit letters, cijfers en onderstrepingstekens mag bestaan, maar met een letter moet beginnen;
- niet dezelfde mag zijn als die van een andere database.

U kunt een expliciet pad van maximaal 64 tekens voor het besturingssysteem opgeven (gebruik geen spaties in de specificatie van het pad). De naam en het pad van de database-bestandsindex worden als rij-invoer in de hoofdcatalogustabel (Sysdbs) opgeslagen.

Als de bij <pad> opgegeven database-bestandsindex nog niet bestaat, wordt er op de met <pad> opgegeven plaats een nieuwe bestandsindex gemaakt. Wanneer de database-bestandsindex al bestaat, wordt deze door SQL als een nieuwe database gedefinieerd en wordt er een set catalogustabellen naar de bestandsindex gekopieerd. Indien u geen pad opgeeft, wordt de database-bestandsindex als sub-bestandsindex onder de actieve bestandsindex gemaakt.

CREATE DATABASE

WAARSCHUWING
Wanneer u probeert met het commando CREATE DATABASE een database-bestandsindex te maken met dezelfde naam en padspecificaties als de SQLHOME-bestandsindex (bijvoorbeeld C:\DBASE\SQLHOME), kunnen SQL-bestanden worden beschadigd.

Nadat u een SQL-database hebt gemaakt, wordt deze automatisch actief gemaakt. U kunt een database ook actief maken met het commando START DATABASE. Om een bepaalde database automatisch actief te maken zodra u de SQL-modus start of een ingebed SQL-programma uitvoert, moet u dit bestand met de opdracht SQLDATABASE in het bestand Config.db opgeven.

Indien dBASE IV met een wachtwoord is beveiligd, kan een database alleen met DROP worden verwijderd door degene die de betreffende database heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Voorbeelden

Om de database Mijninfo als een sub-bestandsindex onder uw actieve bestandsindex te maken typt u:

```
SQL. CREATE DATABASE Mijninfo;
```

Om de database Mijnapp als een sub-bestandsindex onder een bestaande bestandsindex met de naam C:\DBASE te maken typt u:

```
SQL. CREATE DATABASE C:\Dbase\Mijnapp;
```

Zie ook

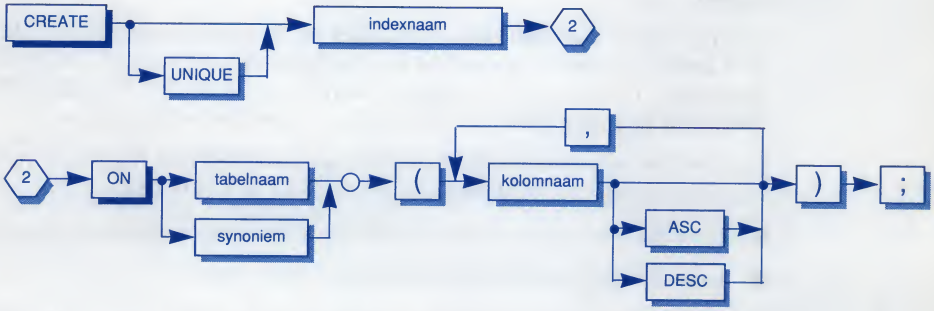
START DATABASE

CREATE INDEX

Met het commando CREATE INDEX maakt u een index op basis van de waarden van één of meer kolommen uit een tabel in de actieve database. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u de tabel zelf hebben gemaakt of over het INDEX-recht voor die tabel beschikken.

Syntaxis

CREATE [UNIQUE] INDEX <indexnaam>
  ON <tabelnaam>
  (<kolomnaam>[ASC/DESC]
  [,<kolomnaam>[ASC/DESC]...]);



Afbeelding 6-5 Het commando CREATE INDEX

Gebruik

Met een index kunt u snel gegevens uit een SQL-tabel ophalen. Met het commando CREATE INDEX kunt u kolommen opgeven. De waarden daarvan in de GROUP BY- en ORDER BY-clausules van commando's voor het bewerken van gegevens worden gebruikt om de uitkomsten te sorteren.

De indexnaam die u bij het commando CREATE INDEX opgeeft, kan een willekeurige in dBASE IV geldige naam zijn. Dat wil zeggen dat de naam:

- uit maximaal tien tekens mag bestaan;
- uit letters, cijfers en onderstrepingstekens mag bestaan, maar met een letter moet beginnen;
- niet dezelfde mag zijn als die van een andere index in dezelfde database, omdat elke indexnaam in een database uniek moet zijn.

Bij het commando CREATE INDEX kunt u de parameter <kolomnaam> gebruiken voor het opgeven van de naam van elke willekeurige indexkolom. U kunt een index opgeven voor kolommen met elk gegevenstype behalve voor die met het gevenstype LOGICAL. Het is mogelijk tot maximaal tien kolommen op te geven voor een index zolang de totale breedte van alle kolommen samen niet meer dan 100 bytes in beslag neemt. Probeert u meer dan tien kolomnamen op te geven, dan verschijnt er een foutmelding.

Met de sleutelwoorden ASC en DESC wordt opgegeven hoe de volgorde van de indexrijen wordt, gebaseerd op de waarden in de geïndexeerde kolom. De volgorde is afdalend respectievelijk oplopend. Aangezien ASC de standaardinstelling is, hoeft u deze parameter niet op te geven wanneer u een indexkolom definieert. U kunt in één index niet zowel oplopende als afdalende sorteervolgorde opgeven.

Door het toevoegen van het sleutelwoord UNIQUE wordt in dBASE IV gecontroleerd of de gegevens in de indexkolom uniek zijn voordat de index wordt gemaakt. Wannneer een unieke index is opgegeven, worden bij de bewerkingen INSERT en UPDATE alle niet unieke waarden voor de indexkolommen geweigerd.

Met CREATE INDEX kunt u maximaal 47 verschillende indexen bij een tabel maken. Een index is gedefinieerd als een label in een .mdx-bestand van dBASE IV met dezelfde naam als de tabel.

In de rijen in een indexlabel staan de waarden van de geïndexeerde kolom en de posities van die waarden in de tabel, gerangschikt in de volgorde die u bij CREATE INDEX hebt opgegeven. Telkens wanneer u tabelrijen bewerkt met INSERT, UPDATE of DELETE moeten ook de corresponderende waarden in de indexlabel worden bijgewerkt.

Tegenover de verbeterde prestaties van het commando SELECT bij gebruik van indexen, staan mogelijke vertragingen bij de bewerkingen INSERT en UPDATE als:

- meerdere indexlabels moeten worden bijgewerkt voor de wijzigingen van een tabel;
- moet worden gecontroleerd of wijzigingen uniek zijn.

U kunt in SQL anders dan in dBASE IV geen specifieke index opgeven voor een bepaalde bewerking. SQL selecteert automatisch de meest efficiënte van de beschikbare indexen.

Indien dBASE IV met een wachtwoord is beveiligd, kan een index met DROP alleen worden verwijderd door degene die de betreffende index heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt. Indexen worden echter automatisch verwijderd wanneer de tabellen waarop deze zijn gebaseerd, worden verwijderd.

OPMERKING

1. *Indexen kunnen alleen voor tabellen en niet voor visies worden gedefinieerd. Wanneer u een visie gebruikt, maakt SQL gebruik van indexen die zijn gedefinieerd voor tabellen die aan die visie ten grondslag liggen.*
2. *Indexrijen bij een kolom van het type CHAR worden gerangschikt op basis van de ASCII-waarden van de kolom. Deze zijn verschillend voor hoofdletters en kleine letters.*
3. *U kunt de gegevens over een index bekijken door deze op te vragen uit de catalogustabel Sysidxs (bijvoorbeeld: SELECT * FROM Sysidxs WHERE Ixname = "Achtnaam" ;). U kunt de indexen die voor tabellen beschikbaar zijn, bekijken door de kolommen van de catalogustabel Sysidxs weer te geven.*

WAARSCHUWING
Database-bestanden van SQL-tabellen met unieke indexen zijn in de dBASE-modus slechts als bestanden met leesrecht beschikbaar. Wilt u zulke bestanden bewerken of er nieuwe gegevens aan toevoegen, dan moet u deze eerst naar een ander dBASE-bestand exporteren met het commando UNLOAD of met de SAVE TO TEMP-clausule van het commando SELECT.

Voorbeelden

Om een index in oplopende volgorde te maken op basis van de kolom Achtnaam van de tabel Klanten, typt u:

```
SQL. CREATE INDEX Achtnaam
     ON Klanten (Achtnaam);
```

Om een index in afdalende volgorde te maken op basis van de kolom Eenhpr van de tabel Inventar, typt u:

```
SQL. CREATE INDEX Kosten
     ON Inventar (Eenhpr DESC);
```

Om een unieke index te maken op basis van de kolom Order_nr van de tabel Verkoop, typt u:

```
SQL. CREATE UNIQUE INDEX Opdr_nr
     ON Verkoop (Order_nr);
```

Zie ook

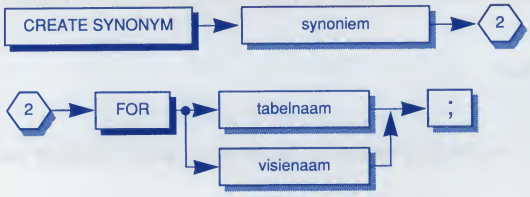
DROP INDEX

CREATE SYNONYM

Met het commando CREATE SYNONYM definieert u een synoniem of alternatieve naam voor een tabel of visie in de actieve database. Om een synoniem te maken hebt u geen speciale rechten nodig. Wanneer dBASE IV echter met een wachtwoord is beveiligd, kunt u een synoniem alleen gebruiken om bewerkingen uit te voeren op tabellen of visies waarvoor u rechten hebt.

Syntaxis

CREATE SYNONYM <synoniem>
FOR <tabelnaam/visienaam>;



Afbeelding 6-6 Het commando CREATE SYNONYM

Gebruik

De alternatieve naam die is gedefinieerd met CREATE SYNONYM kan de actuele naam van een tabel of visie vervangen in alle SQL-commando's met uitzondering van de volgende: CREATE TABLE, DROP TABLE, CREATE VIEW, DROP VIEW. U kunt met het commando CREATE SYNONYM bijvoorbeeld een afkorting definiëren voor een tabel of visie met een onhandig lange naam.

Het synoniem dat u bij het commando CREATE SYNONYM opgeeft, kan een willekeurige in dBASE IV geldige naam zijn. Dat wil zeggen dat het synoniem:

- uit maximaal acht tekens mag bestaan;
- uit letters, cijfers en onderstrepingstekens mag bestaan, maar met een letter moet beginnen;
- niet hetzelfde mag zijn als een ander synoniem en niet de naam van een andere tabel of visie in dezelfde database mag zijn.

Indien dBASE IV met een wachtwoord is beveiligd, kan een synoniem alleen met DROP worden verwijderd door degene die het betreffende synoniem heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt. Synoniemen worden echter automatisch verwijderd wanneer de tabellen of visies waarop zij zijn gebaseerd, worden verwijderd.

OPMERKING
*U kunt gegevens over elk synoniem bekijken door deze op te vragen uit de catalogustabel Syssyns (bijvoorbeeld: SELECT * FROM Syssyns WHERE Syname = "K1" ;).*

Voorbeelden

Om K1 als synoniem voor de tabel Klanten te definiëren typt u:

```
SQL. CREATE SYNONYM K1 FOR Klanten;
```

Om het synoniem vervolgens te gebruiken typt u:

```
SQL. SELECT * FROM K1;
```

Zie ook

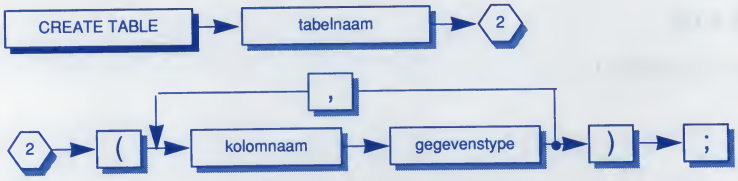
DROP SYNONYM

CREATE TABLE

Met het commando CREATE TABLE kunt u een nieuwe SQL-tabel in de actieve database definiëren. Om een SQL-tabel te maken hebt u geen speciale rechten nodig. Wanneer dBASE IV met een wachtwoord is beveiligd, wordt een tabel gecodeerd zodra deze is gemaakt.

Syntaxis

CREATE TABLE <tabelnaam>
(<kolomnaam><gegevenstype>
[,<kolomnaam><gegevenstype>...]);



Afbeelding 6-7 Het commando CREATE TABLE

Gebruik

Met het commando CREATE TABLE kunt u de naam van een nieuwe tabel en de namen, gegevenstypen en afmetingen van de kolommen in die tabel opgeven. Met de parameter <tabelnaam> geeft u de naam op van de tabel die u wilt maken. Deze naam kan een willekeurige in dBASE geldige naam zijn. Dat wil zeggen dat de naam:

- uit maximaal acht tekens mag bestaan;
- uit letters, cijfers en onderstrepingstekens mag bestaan, maar met een letter moet beginnen;
- niet uit slechts één letter mag bestaan als die tussen A en J ligt;
- niet dezelfde mag zijn als die van een andere tabel of visie en niet als synoniem in gebruik mag zijn.

Wanneer u een tabel maakt, wordt er door SQL een dBASE-database-bestand (.dbf) met dezelfde naam als de tabel gemaakt. Bestaat er al een database-bestand met die naam, dan wordt u ervoor gewaarschuwd dat het bestaande bestand door het nieuwe zal worden overschreven. Velden in het database-bestand corresponderen met de kolommen die voor de tabel zijn opgegeven.

Als SET CATALOG op ON is ingesteld, wordt de nieuwe tabel aan het geopende catalogusbestand toegevoegd.

De gegevenstypen die u kunt gebruiken bij het definiëren van kolommen staan in tabel 6-3.

Tabel 6-3 SQL-gegevenstypen

Gegevenstype	Omschrijving
SMALLINT	Bevat een geheel getal van maximaal zes cijfers (inclusief teken). Ingevoerde waarden liggen tussen -99.999 en 999.999 (wanneer geen plusteken wordt opgegeven). Kolommen met dit gegevenstype komen overeen met velden van het type Numeriek(6,0) uit het database-bestand.
INTEGER	Bevat een geheel getal van maximaal elf cijfers (inclusief teken). Ingevoerde waarden liggen tussen -99.999 en 999.999 (als geen plusteken wordt opgegeven). Kolommen met dit gegevenstype komen overeen met velden van het type Numeriek(11,0) uit het database-bestand.
DECIMAL(x,y)	Bevat een getal met een vast decimaalteken en een teken. Het getal bestaat uit *x* cijfers (inclusief teken) en *y* decimaalposities (significante cijfers rechts van het decimaalteken). *x* heeft een bereik van 0-19 en *y* heeft een bereik van 0-18. Kolommen met dit gegevenstype komen overeen met velden van het type Numeriek(x + 1,y) uit het database-bestand.
NUMERIC(x,y)	Bevat een getal met een vast decimaalteken en een teken. Het getal bestaat uit *x* cijfers (inclusief teken en decimaalteken) en *y* decimaalposities (significante cijfers rechts van het decimaalteken). *x* heeft een bereik van 1-20 en *y* heeft een bereik van 0-18. Kolommen met dit gegevenstype komen overeen met velden van het type Numeriek(x,y) uit het database-bestand.
FLOAT(x,y)	Bevat een getal met een zwevend decimaalteken en een teken. Het getal bestaat uit x cijfers (inclusief teken en decimaalteken) en *y* decimaalposities (significante cijfers rechts van het decimaalteken). *x* heeft een bereik van 1-20 en *y* heeft een bereik van 0-18. Het bereik van de getallen die u kunt opslaan ligt tussen 0,1 10-307 en 0,9 10+308. Een getal kan worden opgegeven in de wetenschappelijke (exponentiële) notatie, bijvoorbeeld -9.99E+235. Kolommen met dit gegevenstype komen overeen met velden van het type Zwevend (x,y) uit het database-bestand.
CHAR(n)	Bevat een tekenreeks van maximaal *n* tekens. *n* heeft een bereik van 1-254. U kunt in een kolom van het type CHAR tekenreeksen of waarden uit een andere kolom van het type CHAR of uit een geheugenvariabele van het type CHAR, invoeren. Kolommen met dit gegevenstype komen overeen met tekenvelden uit het database-bestand.

Tabel 6-3 SQL-gegevenstypen (vervolg)

Gegevenstype	Omschrijving
DATE	Bevat een datum volgens de indeling die bij SET DATE en SET CENTURY is opgegeven. De standaardindeling is *dd/mm/jj*. U kunt in een kolom van het type DATE waarden uit een andere kolom van het type DATE, uit geheugenvariabelen van het type DATE invoeren. U kunt ook datumreeksen die met CTOD zijn geconverteerd (bijvoorbeeld CTOD("15/02/86") of datumreeksen die door { } zijn gescheiden (bijvoorbeeld {15/02/86}), invoeren. Kolommen met dit gegevenstype komen overeen met datumvelden uit het database-bestand.
LOGICAL	Bevat een waarde voor logisch waar of onwaar (.T. staat voor waar, .F. voor onwaar). U kunt in een kolom van het type LOGICAL behalve waarden uit een andere kolom van het type LOGICAL of uit een geheugenvariabele van het type LOGICAL ook de constanten .T., .t., .Y., .y., .F., .f., .J., .j., .N. en .n. invoeren. Kolommen met dit gegevenstype komen overeen met logische velden uit het database-bestand.

OPMERKING

1. *Het totale aantal kolommen in een tabel mag niet groter zijn dan 255 en de totale breedte van alle kolommen samen in een tabel mag niet meer dan 4000 bytes in beslag nemen.*
2. *U kunt in dBASE IV SQL geen dBASE IV-memovelden gebruiken of maken.*
3. *Een kolomnaam kan uit maximaal tien tekens bestaan en mag letters, cijfers en onderstrepingstekens bevatten. In een tabelnaam mogen geen blanco posities voorkomen en de naam moet met een letter beginnen.*

Voorbeeld

Om een tabel met de naam Zending te definiëren waarmee alle zendingen die de firma verlaten, kunnen worden opgeslagen, typt u:

```
SQL. CREATE TABLE Zending
     (Zend_nr    CHAR(6),
     Zend_dat    DATE,
     Order_nr    CHAR(6),
     Exped       CHAR(25),
     Gewicht     DECIMAL(4,1),
     Waarde      DECIMAL(10,2));
```

Zie ook

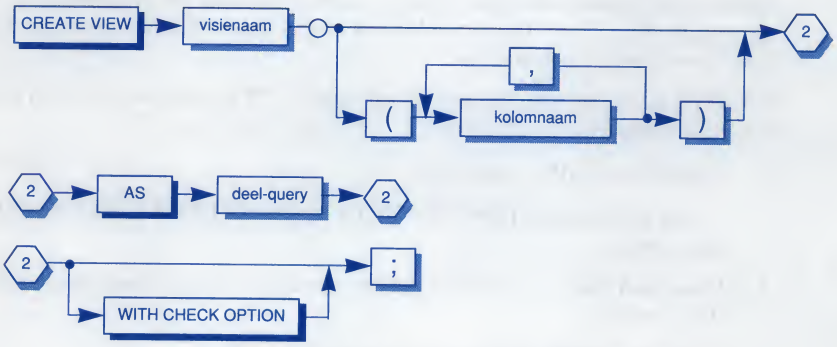
ALTER TABLE, DBDEFINE, DROP TABLE, INSERT, LOAD DATA

CREATE VIEW

Met het commando CREATE VIEW wordt een *virtuele* tabel gemaakt bestaande uit kolommen afkomstig uit één of meer basistabellen (inclusief catalogustabellen) of visies. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u degene zijn die alle tabellen die voor het maken van de visie worden gebruikt, heeft gemaakt, of moet u voor al die tabellen het SELECT-recht hebben.

Syntaxis

CREATE VIEW <visienaam>[(<kolomnaam>, <kolomnaam>...)]
AS <SELECT-commando>
[WITH CHECK OPTION];



Afbeelding 6-8 Het commando CREATE VIEW

Gebruik

De naam voor de visie die u bij CREATE VIEW opgeeft, kan een willekeurige in dBASE geldige naam zijn. Dat wil zeggen dat de naam:

- uit maximaal acht tekens mag bestaan;
- uit letters, cijfers en onderstrepingstekens mag bestaan, maar met een letter moet beginnen;
- niet uit slechts één letter mag bestaan als die tussen A en J ligt;
- niet dezelfde mag zijn als die van een andere visie of tabel en niet als synoniem in gebruik mag zijn. (U mag aan een visie wel de naam van een bestaande index geven, maar dit wordt afgeraden.)

De definitie van een visie wordt opgeslagen in de SQL-catalogustabel Sysviews.

Een visie wordt een virtuele tabel genoemd omdat er niet daadwerkelijk gegevens in worden opgeslagen. In plaats daarvan bevat een visie de *omschrijving*< van de gegevens zoals die wordt afgeleid uit één of meer basistabellen of visies. Wanneer een visie op een andere visie is gebaseerd, worden de gegevens afgeleid uit de basistabellen van die andere visie.

CREATE VIEW

U kunt met SELECT gegevens over visies op dezelfde manier opvragen als bij tabellen. Zodra gegevens in de basistabellen van een visie worden gewijzigd, worden ook de gegevens die bij de visie worden weergegeven, gewijzigd.

Een eenmaal gemaakte visie kan alleen worden verwijderd door degene die de visie heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt. De visie wordt echter automatisch verwijderd wanneer de basistabellen of visies waarop de visie is gebaseerd, worden verwijderd.

Gegevens voor de visie definiëren

Het commando CREATE VIEW initieert een SELECT-commando om kolomwaarden uit de basistabellen en visies op te halen. Gegevens uit kolommen van de basistabel of visie die met de clausule AS <SELECT-commando> zijn opgegeven, worden overgebracht naar de corresponderende visiekolommen.

U kunt bij het maken van een visie elk geldig SELECT-commando geven, zij het met de volgende restricties:

1. U kunt geen UNION opgeven.
2. U kunt geen clausule FOR UPDATE OF, INTO, ORDER BY of SAVE TO TEMP gebruiken.
3. U kunt een visie niet samenvoegen met een andere visie of tabel als u de clausule GROUP BY gebruikt.

Kolommen in een visie benoemen

Met de optionele parameter <kolomnaam> kunt u kolomnamen voor de visie opgeven. Als u dit niet doet, hebben de kolommen in de visie dezelfde naam als in de basistabellen. U moet in ieder geval kolomnamen opgeven als:

- de verwerkte kolommen van de basistabellen constanten, tekenreeksen, geheugenvariabelen, uitdrukkingen of SQL- of dBASE-functies bevatten;
- kolomnamen in twee of meer basistabellen identiek zijn.

OPMERKING
De conventies voor het benoemen van visiekolommen zijn dezelfde als die voor tabelkolommen (zie de derde opmerking bij CREATE TABLE). De afmetingseisen (aantal rijen en kolommen) voor een visie zijn gelijk aan die voor tabellen (zie de eerste opmerking bij CREATE TABLE).

Gegevens opvragen uit een visie

U kunt elk geldig SELECT-commando gebruiken om gegevens uit een visie op te vragen. Wanneer een SELECT-commando dat geldig is voor het opvragen van gegevens uit een tabel, echter wordt uitgebreid voor de visiedefinitie, is het commando mogelijk niet meer geldig voor de visie.

Bekijk het volgende commando CREATE VIEW:

```
CREATE VIEW Medew1 (C1, C2)
    AS SELECT AVG(Salaris), Standpl
    FROM Medew
    GROUP BY Standpl;
```

Het opvragen met:

```
SELECT C1 FROM Medew1;
```

is ongeldig voor de visie Medew1, aangezien in de uitbreiding:

```
SELECT AVG(Salaris) FROM Medew
GROUP BY Standpl;
```

de kolom GROUP BY (Standpl) niet in de SELECT-clausule is opgenomen. Met de volgende commando's kunnen wel gegevens worden opgevraagd uit de visie Medew1:

```
SELECT * FROM Medew1;
```

```
SELECT C2 FROM Medew1;
```

Een bewerkbare visie gebruiken

Wanneer een visie *bewerkbaar* is, kunt u de basistabel bijwerken door gegevens in de visie bij te werken, aan de visie toe te voegen, of uit de visie te verwijderen. Een visie is bewerkbaar wanneer:

- deze op slechts één basistabel is gebaseerd;
- deze geen self-join met de basistabel definieert;
- deze op een bewerkbare visie is gebaseerd.

Als u een bewerkbare visie wilt maken, moet u over het recht beschikken om met INSERT, UPDATE of DELETE de basistabel of bewerkbare visie te wijzigen.

Een visie is niet bewerkbaar indien het voor de definitie gebruikte SELECT-commando:

- in de SELECT-clausule het sleutelwoord DISTINCT, een SQL-verzamelfunctie of een dBASE-functie bevat;
- een FROM-clausule bevat met een niet bewerkbare visie of met meer dan één tabel of visie;
- een GROUP BY-clausule bevat;
- een geneste sub-query bevat met een FROM-clausule die verwijst naar de basistabel van de visie.

CREATE VIEW

Een visiekolom die is afgeleid door een rekenkundige bewerking of constante te gebruiken, kan niet worden bijgewerkt.

Als u met het commando CREATE VIEW een bewerkbare visie maakt kunt u de parameter WITH CHECK OPTION opgeven. Daardoor worden rijen die tijdens het gebruik van de visie worden toegevoegd, bijgewerkt of verwijderd, vergeleken met de WHERE-voorwaarde uit het SELECT-commando van de visie. Wanneer een rij niet voldoet aan de voorwaarde, zal de basistabel niet worden bijgewerkt met deze rij.

Voorbeelden

Om een visie met alle kolommen uit de tabel Verkopen te maken waarin alleen de rijen met niet gefactureerde orders staan typt u:

```
SQL. CREATE VIEW Factuur
     AS SELECT*
     FROM Verkopen
     WHERE NOT Gefact;
```

Om een bewerkbare visie met een set kolommen uit de tabel Medew te maken typt u:

```
SQL. CREATE VIEW Ad
     AS SELECT Medew_nr, Achtnaam, Voornaam, Indienst
     FROM Medew
     WHERE Standpl = "Amsterdam";
```

Om een visie met een combinatie van de kolommen Achtnaam en Voornaam uit de tabel Klanten te maken typt u:

```
SQL. CREATE VIEW Adres
     (Helenaam, Plaats, Prov)
     AS SELECT Voornaam+Achtnaam, Plaats, Prov
     FROM Klanten;
```

Om een visie met een combinatie van gegevens uit kolommen van twee verschillende tabellen te maken, typt u:

```
SQL. CREATE VIEW Order
     (Order_nr, Verk_dat, Verkoper)
     AS SELECT Verkopen.Order_nr, Verk_dat,
     FROM Verkopen, Medew
     WHERE Verkopen.Medewnr = Medew.Medewnr;
```

Om tijdens het gebruik van de visie Ad de spelling van de voornaam van Willemse in de tabel Medew te wijzigen typt u:

```
SQL. UPDATE Ad
        SET Voornaam = "Paul"
        WHERE Achternaam = "Willemse";
```

Zie ook

DROP VIEW, SELECT

DBCHECK

Met het commando DBCHECK controleert u voor een bepaalde tabel of voor alle tabellen in de actieve database of de ingave in de catalogustabel overeenkomt met de bestandsstructuur van de .dbf- en .mdx-bestanden die aan de tabel ten grondslag liggen.

Syntaxis

DBCHECK [<tabelnaam>];

Afbeelding 6-9 Het commando DBCHECK

Gebruik

Wanneer u het commando DBCHECK zonder de parameter <tabelnaam> gebruikt, worden de bestanden bij alle SQL-tabellen en de bijbehorende indexen in de actieve database gecontroleerd. Als u een tabelnaam opgeeft, worden alleen de .dbf- en .mdx-bestanden van die tabel gecontroleerd.

Voor elke tabel waarvoor de ingave in de catalogustabel van de SQL-database niet overeenkomt met de bestandsstructuur van de bij de tabel behorende .dbf- en .mdx-bestanden, wordt een foutmelding weergegeven. Wanneer er een foutmelding verschijnt, moet u de volgende procedure volgen:

1. Kopieer de in de foutmeling vermelde .dbf- en .mdx-bestanden van de actieve database-bestandsindex naar een andere bestandsindex of een diskette.
2. Verwijder in de SQL-modus met DROP alle tabellen die in de foutmelding worden vermeld, zodat de ingaven voor die tabellen uit de catalogustabel worden verwijderd.
3. Kopieer de .dbf- en.mdx-bestanden terug naar de database-bestandsindex.
4. Geef in de SQL-modus het commando DBDEFINE voor ieder .dbf-bestand dat u opnieuw als SQL-tabel wilt definiëren. De bijbehorende indexen worden bij de bewerking DBDEFINE automatisch opnieuw gedefinieerd.

Wanneer u het commando DBCHECK uitvoert voor een tabel in de actieve database terwijl het corresponderende database-bestand in de dBASE-modus is geopend, verschijnt er een foutmelding.

OPMERKING
U kunt met DBCHECK geen bestanden controleren die in de dBASE-modus met het commando PROTECT zijn gecodeerd.

Voorbeelden

Om de ingaven in de catalogustabel voor de tabel Medew te controleren typt u:

```
SQL. DBCHECK Medew;
```

Om de ingaven in de catalogustabel voor alle tabellen uit de database Voorbld te controleren typt u:

```
SQL. DBCHECK;
```

Zie ook

DBDEFINE

DBDEFINE

Met het commando DBDEFINE worden ingaven voor de catalogustabel gemaakt van één of meer database-bestanden en bijbehorende samengestelde index-bestanden in de actieve database. U hoeft niet over speciale rechten te beschikken om DBDEFINE uit te voeren. Wanneer dBASE IV met een wachtwoord is beveiligd, worden nieuwe tabellen tijdens de uitvoering van DBDEFINE door SQL gecodeerd en wordt de actieve gebruikersnaam beschouwd als de naam van degene die de tabellen heeft gemaakt.

Syntaxis

DBDEFINE [<.dbf-bestandsnaam>];

Afbeelding 6-10 Het commando DBDEFINE

Gebruik

Wanneer u het commando DBDEFINE zonder de parameter <.dbf-bestandsnaam> geeft, wordt voor elk .dbf-bestand en bijbehorend .mdx-bestand in de actieve SQL-database dat nog niet als tabel of als index is gedefinieerd, een catalogusingave gemaakt. Als u bij het commando DBDEFINE een bestandsnaam opgeeft, worden alleen voor het genoemde .dbf-bestand en de bijbehorende .mdx-bestanden catalogusingaven gemaakt.

OPMERKING
U kunt er op twee manieren voor zorgen dat de catalogusingave voor een index na uitvoering van DBDEFINE overeenkomt met de stand van zaken in het corresponderende database-bestand. U kunt de bestanden die u met DBDEFINE wilt bewerken, eerst met het commando REINDEX opnieuw indexeren voordat u de SQL-modus start. U kunt ook na het commando DBDEFINE het SQL-commando RUNSTATS uitvoeren.

U kunt met DBDEFINE geen een catalogustabelingaven maken voor:

- een .ndx-indexbestand (u moet het bestand dan eerst converteren naar een .mdx-label);
- een .mdx-label waarvan de indexsleutel zou resulteren in een SQL-index van meer dan tien kolommen;
- een UNIQUE-index die in de dBASE-modus is gemaakt;
- memovelden in een database-bestand (in de SQL-modus zijn memovelden niet toegankelijk);
- visies van dBASE (nadat u de basistabellen van een visie met DBDEFINE hebt bewerkt, moet u de visie in de SQL-modus met het commando CREATE VIEW opnieuw maken);

- een database-bestand dat in dBASE-modus is gecodeerd. Om het bestand te decoderen voordat u de SQL-modus start en DBDEFINE uitvoert, moet u SET ENCRYPTION op OFF instellen en met het commando COPY TO een niet gecodeerde versie van het bestand maken. U moet over de juiste PROTECT-rechten beschikken om toegang tot het originele bestand te hebben.
- een database bestand dat in de SQL-modus is gecodeerd. Gebruik het commando UNLOAD DATA om een gedecodeerde versie van het bestand te maken voordat u DBDEFINE uitvoert.
- een database-bestand dat in dBASE-modus is geopend (als u dan DBDEFINE uitvoert, verschijnt een melding waarin staat dat het bestand al is geopend). Gebruik het commando CLOSE om het bestand te sluiten voordat u het commando DBDEFINE gebruikt.
- een bestand waarvan de naam uit één enkele letter bestaat die tussen A en J ligt.

Het commando DBDEFINE geeft foutmeldingen bij bestanden die niet kunnen worden gelezen, waarvoor geen tabellen of indexen kunnen worden gemaakt, of waarvoor de catalogusingaven al bestaan.

Hoewel memovelden in de SQL-modus niet toegankelijk zijn, moeten bijbehorende memobestanden (.dbt-bestanden) in de database-bestandsindex beschikbaar zijn wanneer het database-bestand waartoe zij behoren, met DBDEFINE wordt bewerkt. Als het memobestand niet beschikbaar is, wordt er een foutmelding weergegeven. In dat geval moet u voordat u het commando DBDEFINE geeft, een van de volgende handelingen uitvoeren:

- Kopieer de memobestanden naar de database-bestandsindex.
- Geef bij het database-bestand het commando USE, waardoor u lege memovelden maakt.
- Gebruik het commando MODIFY STRUCTURE om het ontwerpscherm van de database weer te geven en verwijder vervolgens het memoveld.

Voorbeelden

Om catalogusingaven voor het bestand Detail te maken, typt u:

```
SQL. DBDEFINE Detail;
```

Om catalogusingaven te maken voor alle database-bestanden die in de actieve database-bestandsindex niet als tabel zijn gedefinieerd, typt u:

```
SQL. DBDEFINE;
```

Zie ook

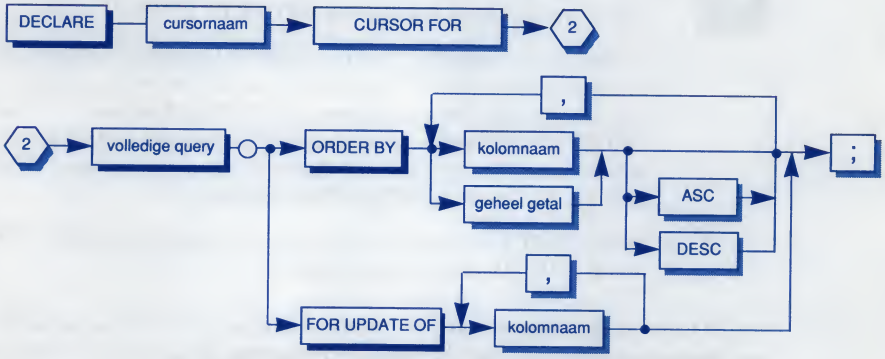
CREATE DATABASE, DBCHECK, RUNSTATS, START DATABASE

DECLARE CURSOR

Met het commando DECLARE CURSOR wordt een cursor gedefinieerd en een bijbehorend SELECT-commando geïnitieerd. Met de cursor kan in een programma in de ingebedde SQL-modus het SELECT-commando worden uitgevoerd waarna de rijen uit de resultatentabel één voor één kunnen worden verwerkt. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u over rechten beschikken voor alle objecten die in het SELECT-commando worden opgenomen.

Syntaxis

DECLARE <cursornaam> CURSOR
  FOR <SELECT-commando>
  [FOR UPDATE OF <kolommenlijst>/<ORDER BY-clausule>;



Afbeelding 6-11 Het commando DECLARE CURSOR

Gebruik

In een ingebed SQL-programma wordt een cursor gebruikt om één voor één de rijen in een resultatentabel aan te wijzen. De gegevens in de rij waarnaar de cursor wijst, worden voor verwerking naar dBASE-geheugenvariabelen overgebracht.

In het programma wordt eerst het SELECT-commando van de cursor uitgevoerd, door met het commando OPEN <cursor> een resultatentabel te maken. Door het commando FETCH wordt de cursor achtereenvolgens naar elk van de rijen van de resultatentabel verplaatst, waarna de gegevens in de rijen worden overgebracht naar de bij FETCH opgegeven geheugenvariabelen.

Wanneer een cursor is gedefinieerd en geopend, blijft deze gedefinieerd en geopend totdat:

- het programma het commando CLOSE <cursor> uitvoert, de SQL-modus verlaat of geheel is uitgevoerd (de cursor blijft dus geopend tijdens de uitvoering van subroutines van een lager niveau, ongeacht of het programmabestanden van dBASE (.prg) of van SQL (.prs) betreft).
- de database waarin de cursor is geopend, wordt gesloten met het commando STOP DATABASE of DROP DATABASE, of doordat met START DATABASE een andere database wordt geopend dan wel met CREATE DATABASE een nieuwe database wordt gemaakt.

DECLARE CURSOR

Met een cursor kunnen in een programma rijen uit de basistabel die corresponderen met de rijen uit de resultatentabel van SELECT, worden verwijderd of bijgewerkt.

- Het SELECT-commando heeft de clausule FOR UPDATE OF. Daarmee is het mogelijk om met het commando UPDATE...WHERE CURRENT OF waarden in de rij uit de basistabel bij te werken die correspondeert met de rij in de resultatentabel waarnaar de cursor wijst.
- Met de clausule WHERE CURRENT OF van het commando DELETE is het mogelijk in een programma de rij uit de basistabel te verwijderen die correspondeert met de rij uit de resultatentabel waarnaar de cursor wijst.

OPMERKING

1. *In een SELECT-commando waarin een clausule FOR UPDATE OF staat, kan niettegelijkertijd een clausule ORDER BY of UNION staan.*
2. *U kunt met de clausule UPDATE...WHERE CURRENT OF geen rijen bijwerken als het SELECT-commando van de cursor zogenaamde verzamelfuncties, het sleutelwoord DISTINCT of één van de clausules GROUP BY, HAVING of UNION bevat. De clausule FROM kan niet meer bevatten dan één tabel, een niet-bewerkbare visie of een tabel die ook in een sub-query in het commando SELECT is opgegeven*
3. *In het SELECT-commando van een cursor kunnen de clausules INTO of SAVE TOTEMP niet zijn opgenomen.*

Een SQL-programmabestand kan niet verwijzen naar een cursor die in een ander SQL-programmabestand is gedefinieerd. Wanneer procedures uit een programmabestand verwijzen naar een cursor die in een andere procedure in hetzelfde bestand is gedefinieerd, moet de procedure waarin de cursor wordt gedefinieerd, fysiek voorafgaan aan de procedures die naar die cursor verwijzen.

U kunt het commando DECLARE CURSOR niet voorwaardelijk uitvoeren met de programmaconstructie IF...ENDIF.

Voorbeelden

Om een cursor te definiëren waarmee rijen uit de tabel Verkopen worden weergegeven indien een order niet is gefactureerd, kunt u het volgende commando in een programma opnemen:

```
DECLARE Fact CURSOR FOR
    SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
    FROM Verkopen;
    WHERE NOT Gefact;
```

In het volgende voorbeeld wordt deze cursor uitgebreid waardoor de rijen in de tabel Verkopen kunnen worden bijgewerkt.

```
DECLARE Fact CURSOR FOR
  SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
  FROM Verkopen
  WHERE NOT Gefact;
  FOR UPDATE OF Verkopen
  .
  .
  .
OPEN Fact;
 DO WHILE .T.
  FETCH Fact INTO morder_nr, mverk_dat, mmedew_nr, mklant_nr, mgefact;
  IF SQLCODE = 0
      * Druk een factuur voor ordernummer in de huidige rij af
      * Indien gelukt, wijzig geheugenvariabele mgefact in .T.
      *
    DO Factuur WITH morder_nr, mverk_dat, mmedew_nr, mklant_nr, ; mgefact
      *
    IF mgefact
          UPDATE Verkopen
          SET Gefact = .T.
          WHERE CURRENT OF Fact;
          *
      ENDIF
  ELSE
    EXIT
  ENDIF
 ENDDO
CLOSE Fact;
```

Zie ook

CLOSE, FETCH, OPEN

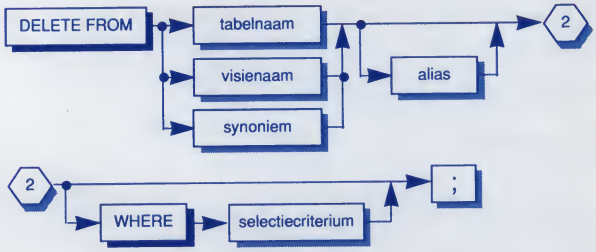
DELETE

Met het commando DELETE kunt u opgegeven rijen uit een tabel verwijderen. Wanneer dBASE met een wachtwoord is beveiligd, dient u te beschikken over het DELETE-recht voor de tabel of voor de bewerkbare visie waarmee u toegang tot de tabel hebt.

Het commando DELETE kent twee vormen. De eerste vorm, doorgaans met een WHERE-clausule, kan in de interactieve of in de ingebedde SQL-modus worden uitgevoerd. De tweede vorm, waarin de clausule WHERE CURRENT OF is opgenomen, wordt uitsluitend in de ingebedde SQL-modus met een gedefinieerde cursor gebruikt.

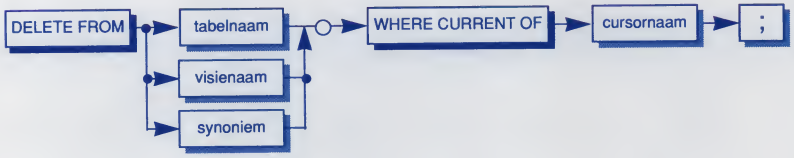
Syntaxis

DELETE
    FROM <tabelnaam> [<alias-naam>]
    [WHERE-clausule];



Afbeelding 6-12 Het commando DELETE; de vorm DELETE...WHERE

DELETE
    FROM <tabelnaam>
    WHERE CURRENT OF <cursornaam>;



Afbeelding 6-13 Het commando DELETE; de vorm DELETE... WHERE CURRENT OF

Gebruik

Wanneer u met behulp van een visie rijen uit een basistabel wilt verwijderen, moet die visie bewerkbaar zijn (raadpleeg het gedeelte over het commando CREATE VIEW).

DELETE met de clausule WHERE

Deze variant van het commando DELETE gebruikt een clausule WHERE voor het selecteren van de rijen die moeten worden verwijderd. In de clausule WHERE kunt u enkelvoudige zoekvoorwaarden, gecombineerde zoekvoorwaarden en enkelvoudige (niet-gecorreleerde) sub-queries opnemen. Een WHERE-sub-query kan niet verwijzen naar de tabel of visie waaruit u rijen verwijdert.

TIP
Om te controleren of met de opgegeven clausule WHERE de juiste rijen worden verwijderd, kunt u het SELECT-commando met dezelfde WHERE-clausule uitvoeren.

WAARSCHUWING
Anders dan bij het dBASE IV-commando DELETE, waarmee slechts wismarkeringen worden aangebracht, worden de rijen bij het SQL-commando DELETE direct uit de tabel verwijderd. Wanneer u DELETE uitvoert zonder de WHERE-clausule te gebruiken en SET SAFETY op ON is ingesteld, wordt er een waarschuwing weergegeven. Wanneer u er voor kiest het commando uit te voeren, worden alle rijen uit de tabel verwijderd.

Als het commando DELETE in een transactie (door het commando BEGIN/END TRANSACTION te gebruiken) als onderdeel van een programma wordt uitgevoerd, worden verwijderde rijen niet direct uit de tabel verwijderd. Dergelijke rijen worden nog in resultatentabellen weergeven (voorafgegaan door een asterisk) totdat de transactie is beëindigd.

U kunt het dBASE-commando SET DELETED op ON instellen, waarmee u voorkomt dat de verwijderde rijen tijdens de uitvoering van de transactie worden weergegeven of verwerkt. Dit is met name belangrijk wanneer andere gebruikers in een netwerk tegelijkertijd dezelfde gegevens bekijken.

DELETE met de clausule WHERE CURRENT OF

De tweede variant van het commando wordt alleen in een ingebed SQL-programma gebruikt. Met DELETE...WHERE CURRENT OF worden de rijen uit een basistabel verwijderd die overeenkomen met de rijen in de resultatentabel waarnaar de cursor wijst.

In het programma moeten de volgende commando's in de opgegeven volgorde zijn opgenomen:

1. DECLARE CURSOR - Definieert de cursor waarnaar vervolgens in de clausule WHERE CURRENT OF wordt verwezen; initieert een SELECT-commando waarmee wordt verwezen naar de tabel of bewerkbare visie waaruit rijen moeten worden verwijderd.
2. OPEN - Voert het SELECT-commando van de cursor uit en maakt zo de resultatentabel voor de rijen die moeten worden verwijderd.
3. FETCH - Plaatst de cursor op de rijen in de resultatentabel; kopieert waarden uit de rijen naar dBASE-geheugenvariabelen.

4. DELETE...WHERE CURRENT OF - Verwijdert (afhankelijk van de met FETCH gekopieerde waarden van de geheugenvariabelen) de rij in de basistabel die correspondeert met de rij in de resultatentabel die door de cursor wordt aangewezen.

De programmaconstructie IF...ELSE wordt gebruikt om geheugenvariabelen te toetsen en zo te bepalen of een rij moet worden verwijderd. De IF-voorwaarde is de voorwaarde voor verwijderen (bijvoorbeeld IF geheugenvariabele mgefact op .T. is ingesteld).

Wanneer u het commando DELETE in een programma gebruikt, moet u het in een transactie (met het dBASE-commando BEGIN/END TRANSACTION) opnemen. Op die manier hebt u de mogelijkheid om de tabel met het commando ROLLBACK terug te brengen in de oorspronkelijke staat wanneer de transactie niet met succes is uitgevoerd. Ook waarborgt u zo de betrouwbaarheid van de database voor andere gebruikers in een netwerk.

OPMERKING
Rijen die tijdens een transactie met het commando DELETE...WHERE CURRENT OF worden verwijderd, worden niet uit de tabel verwijderd voor beëindiging van de transactie waarin de bijbehorende cursor wordt gesloten. Als de cursor buiten de transactie wordt geopend en gesloten, worden de rijen niet verwijderd voordat de cursor is gesloten.

Voorbeelden

Om een rij uit de tabel Medew te verwijderen, typt u:

```
SQL. DELETE FROM Medew
     WHERE Achtnaam = "Willemse";
```

De melding 1 rij(en) gewist wordt weergegeven.

Om alle rijen uit de tabel Verkopen te verwijderen waarin een Verk_dat staat die voor 22 september 1990 valt typt u:

```
SQL. DELETE
     FROM Verkopen
     WHERE Verk_dat < {22/09/90};
```

De melding **4 rij(en) gewist** wordt weergegeven.

In het volgende voorbeeld ziet u hoe het commando DELETE...WHERE CURRENT OF in combinatie met een cursor wordt gebruikt.

```
DECLARE Fact CURSOR FOR
 SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
 FROM Verkoop
 WHERE NOT Gefact;
  .
  .
  .
OPEN Fact;
 DO WHILE .T.
  BEGIN TRANSACTION
  FETCH Fact INTO morder_nr, mverk_dat, mmedew_nr, mklant_nr, mgefact;
  IF SQLCODE = 0
   * Druk een factuur voor ordernummer in de huidige rij af
   * Indien gelukt, wijzig geheugenvariabele mgefact in .T.
   *
   DO Factuur WITH morder_nr, mverk_dat, mmedew_nr, mklant_nr, ; mgefact
   *
   IF mgefact .AND. Verk_dat < {22/09/90})
    DELETE FROM Verkoop
    WHERE CURRENT OF Fact;
    ENDIF
  ELSE
   EXIT
  ENDIF
  END TRANSACTION
 ENDDO
CLOSE Fact;
```

Zie ook

DECLARE CURSOR, FETCH, SELECT

DROP DATABASE

Met het commando DROP DATABASE kunt u een database verwijderen. Indien dBASE IV met een wachtwoord is beveiligd, kan een database met DROP alleen worden verwijderd door degene die de database heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Syntaxis

DROP DATABASE <database-naam>

Afbeelding 6-14 Het commando DROP DATABASE

Gebruik

Het commando DROP DATABASE verwijdert een database door het verwijderen van:

- alle .dbf- en .mdx-bestanden uit de database-bestandsindex waarvoor een corresponderende ingave in de catalogustabellen Systables en Sysidxs is opgenomen;
- de ingave voor de database uit de catalogustabel Sysdbs;
- ingaven voor de tabellen in de database uit het geopende dBASE-catalogusbestand (als SET CATALOG op ON is ingesteld).

Met DROP DATABASE wordt de bestandsindex waarin de database staat, niet verwijderd.

Voordat u een database kunt verwijderen, moet u deze met het commando STOP DATABASE inactief maken. Wanneer u het commando DROP DATABASE uitvoert, verschijnt er een keuzevenster waarin u ervoor gewaarschuwd wordt, dat alle SQL-tabellen in de database zullen worden verwijderd. Door de optie Annuleren te selecteren, kunt u de bewerking afbreken.

Wanneer de database die bij DROP DATABASE wordt opgegeven, niet kan worden gevonden, verschijnt er een foutmelding. Als u er voor kiest de bewerking uit te voeren, worden alle aangetroffen gegevens verwijderd, ook de verwijzingen naar de database die in de hoofdcatalogustabel Sysdbs zijn opgenomen.

OPMERKING
In een netwerk kunt u een database die door een andere gebruiker actief is gemaakt, niet met het commando DROP verwijderen.

Zie ook

CREATE DATABASE, SHOW DATABASE, START DATABASE, STOP DATABASE

DROP INDEX

Met het commando DROP INDEX verwijdert u een index die voor de kolommen van een tabel is gedefinieerd. Indien dBASE IV met een wachtwoord is beveiligd, kan een index met DROP alleen worden verwijderd door degene die de betreffende index heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Syntaxis

DROP INDEX <indexnaam>;

Afbeelding 6-15 Het commando DROP INDEX

Gebruik

Met het commando DROP INDEX kunt u een opgegeven index verwijderen. Indexen worden automatisch verwijderd als de tabellen waarop de indexen zijn gebaseerd, worden verwijderd.

Zie ook

CREATE INDEX, DROP TABLE

DROP SYNONYM

Met het commando DROP SYNONYM kunt u een synoniem (alternatieve naam) voor een tabelnaam of visienaam verwijderen. Indien dBASE IV met een wachtwoord is beveiligd, kan een synoniem met DROP alleen worden verwijderd door degene die het synoniem heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Syntaxis

DROP SYNONYM <synoniem>;

Afbeelding 6-16 Het commando DROP SYNONYM

Gebruik

Met het commando DROP SYNONYM kunt u een opgegeven synoniem verwijderen. Synoniemen worden automatisch verwijderd als de tabel of visie waarvoor het synoniem is opgegeven, wordt verwijderd.

Zie ook

CREATE SYNONYM

DROP TABLE

Met het commando DROP TABLE kunt u een tabel verwijderen. Indien dBASE IV met een wachtwoord is beveiligd, kan een tabel met DROP alleen worden verwijderd door degene die de tabel heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Syntaxis

DROP TABLE <tabelnaam>;

Afbeelding 6-17 Het commando DROP TABLE

Gebruik

Wanneer het commando DROP TABLE wordt uitgevoerd, wordt van een tabel het volgende verwijderd:

- het database-bestand met de gegevens uit de tabel;
- alle verwijzingen naar de tabel die in de SQL-catalogustabellen staan;
- alle indexen, visies en synoniemen die op de tabel zijn gebaseerd;
- de ingave voor de tabel in het geopende dBASE-catalogusbestand (als SET CATALOG op ON is ingesteld).

U kunt in plaats van de parameter <tabelnaam> geen synoniem opgeven.

WAARSCHUWING
Een tabel die met DROP TABLE is verwijderd, kan niet worden hersteld. Wanneer u per ongeluk een tabel hebt verwijderd, moet u deze en de bijbehorende objecten opnieuw *definiëren en opnieuw met gegevens vullen.*

Zie ook

CREATE TABLE, DELETE

DROP VIEW

Met het commando DROP VIEW kunt u een visie verwijderen. Indien dBASE IV met een wachtwoord is beveiligd, kan een visie met DROP alleen worden verwijderd door degene die de tabel waarop de visie is gebaseerd, heeft gemaakt of over de SQLDBA-gebruikersnaam beschikt.

Syntaxis

DROP VIEW <visienaam>;

Afbeelding 6-18 Het commando DROP VIEW

Gebruik

Wanneer het commando DROP VIEW wordt uitgevoerd, wordt het volgende verwijderd:

- de definities van de visie in de SQL-catalogustabellen;
- alle synoniemen en visies die op de visie zijn gebaseerd.

Het verwijderen van een visie heeft geen gevolgen voor de tabellen of visies waarop de visie is gebaseerd.

Een visie en de bijbehorende synoniemen worden automatisch verwijderd als de basistabellen van de visie worden verwijderd.

U kunt in plaats van de parameter <visienaam> geen synoniem opgeven.

Zie ook

CREATE VIEW, DROP TABLE

Met het commando FETCH wordt de cursor achtereenvolgens naar alle rijen van de resultatentabel van een cursor verplaatst, waarna de waarden uit de rijen naar corresponderende dBASE-geheugenvariabelen worden overgebracht. Het commando kan niet interactief worden gebruikt, maar slechts in ingebedde SQL-programma's.

Syntaxis

FETCH <cursornaam>
    INTO <lijst geheugenvariabelen>;

Afbeelding 6-19 Het commando FETCH

Gebruik

Door het commando FETCH op te nemen in een programma kunt u de waarden uit een resultatentabel van een gedefinieerde cursor bewerken. Het commando FETCH werkt als volgt in combinatie met DECLARE CURSOR en OPEN:

1. Met het commando DECLARE CURSOR wordt een cursor gemaakt en het bijbehorende SELECT-commando geïnitieerd.
2. Met het commando OPEN wordt het SELECT-commando uitgevoerd waardoor een resultatentabel wordt gemaakt.
3. Door uitvoering van het commando FETCH worden de geheugenvariabelen uit <lijst geheugenvariabelen> gemaakt. Vervolgens wordt de cursor per rij door de resultatentabel verplaatst, waarna de waarden uit de rijen naar de geheugenvariabelen worden overgebracht.

Elke geheugenvariabele die bij het commando FETCH in de lijst met geheugenvariabelen is opgenomen, moet van links naar rechts overeenkomen met een kolom in de resultatentabel die met SELECT is gemaakt. Wanneer het SELECT-commando van de cursor geen enkele rij oplevert, worden de geheugenvariabelen die bij FETCH zijn opgegeven, niet gemaakt.

Wanneer dBASE IV met een wachtwoord is beveiligd, wordt gecontroleerd of degene die het programma uitvoert, beschikt over rechten voor de tabel die in de FROM-clausule van het SELECT-commando wordt genoemd. Pas daarna wordt het commando DECLARE CURSOR of OPEN uitgevoerd.

Als het commando FETCH voor de eerste keer wordt uitgevoerd, wordt de cursor in de eerste rij van de resultatentabel geplaatst. Vervolgens wordt de cursor telkens wanneer het commando wordt uitgevoerd, naar de volgende rij verplaatst, totdat de laatste rij van de resultatentabel is bereikt.

OPMERKING
De cursor kan alleen voorwaarts worden verplaatst. Als de cursor terug moet worden teruggezet naar de eerste rij zonder alle rijen te doorlopen, moet u de cursor met CLOSE sluiten en met OPEN opnieuw openen.

U kunt in uw programma de SQL-systeemstatusvariabelen Sqlcnt en Sqlcode gebruiken om de status van een FETCH-bewerking vast te stellen. Wanneer door het commando OPEN het SELECT-commando wordt uitgevoerd, bevat Sqlcnt het aantal rijen in de resultatentabel. Als SELECT geen rijen oplevert, bevat Sqlcnt de waarde 0 en Sqlcode de waarde 100.

TIP
Aangezien het SELECT-commando van de cursor soms geen rijen oplevert, moet u ervoor zorgen dat in uw programma de waarde van Sqlcnt wordt gecontroleerd voordat wordt geprobeerd de met FETCH verkregen geheugenvariabelen te verwerken.

Nadat het commando FETCH succesvol de cursor heeft verplaatst en de waarden uit de rijen naar de geheugenvariabelen heeft overgebracht, bevat Sqlcode de waarde 0. Als het commando FETCH niet succesvol is uitgevoerd, bevat Sqlcode de waarde -1. Als de laatste rij is verwerkt, bevat Sqlcode de waarde 100, waarmee wordt aangegeven dat er geen resterende rijen zijn.

Voorbeeld

In het volgende voorbeeld is te zien, hoe de kolommen Order_nr, Verk_dat, Medew_nr, Klant_nr en Gefact uit de tabel Verkopen met het commando FETCH worden verwerkt. De volgende commando's kunnen deel uitmaken van een SQL-programma.

```
DECLARE Fact CURSOR FOR
 SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
 FROM Verkoop
 WHERE NOT Gefact;
  .
  .
  .
OPEN Fact;
  .
  .
  .
IF SQLCODE = 0
  DO WHILE .T.
   FETCH Fact INTO morder_nr, mverk_dat, mmedew_nr, mklant_nr, mgefact;
   IF SQLCNT > 0
     .
     .
     .
    ELSE
     EXIT
    ENDIF
  ENDDO
ENDIF
```

Met de eerste IF-procedure wordt gecontroleerd of de FETCH-bewerking succesvol is geweest, terwijl met de tweede IF-procedure de waarde van Sqlcnt wordt gecontroleerd om te bepalen of er in de resultatentabel nog rijen zijn te verwerken. Als dat niet het geval is, wordt het programma beëindigd.

Zie ook

DECLARE CURSOR, DELETE, FETCH, UPDATE

GRANT

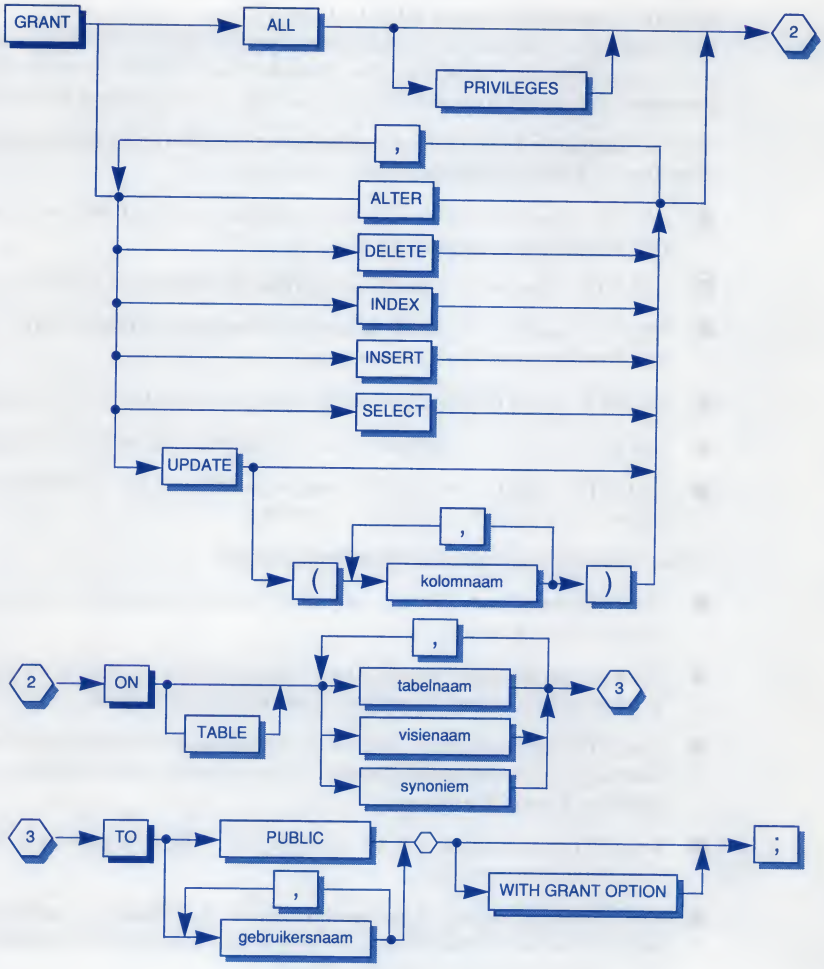
Met het commando GRANT worden toegangsrechten voor tabellen en visies in de actieve database toegewezen aan gebruikersnamen die met het dBASE-commando PROTECT zijn gemaakt.

Syntaxis

```
GRANT ALL [PRIVILEGES]/<rechtenlijst>
    ON [TABLE] <tabellenlijst>
    TO PUBLIC/<gebruikerslijst>
    [WITH GRANT OPTION];
```

De rechtenlijst bestaat uit één of meer van de volgende elementen:

```
ALTER
DELETE
INDEX
INSERT
SELECT
UPDATE [(<kolommenlijst>)]
```



Afbeelding 6-20 Het commando GRANT

Gebruik

In SQL beschikt u over de commando's GRANT en REVOKE om rechten voor tabellen en visies in een database toe te wijzen en in te trekken. Alle rechten die met GRANT en REVOKE worden bestuurd, worden in de SQL-catalogustabel voor de database opgeslagen.

GRANT

De commando's GRANT en REVOKE werken in combinatie met de dBASE IV-beveiliging middels wachtwoorden. Deze wordt met het commando PROTECT door een database-beheerder ingesteld. Wanneer dBASE IV eenmaal met een wachtwoord is beveiligd, moet iedere gebruiker van dBASE IV zich met een gebruikersnaam aanmelden. Als een gebruiker na het aanmelden naar de SQL-modus schakelt, zijn de bewerkingen die hij per database kan uitvoeren, vastgelegd door de rechten die aan zijn gebruikersnaam zijn toegewezen.

Met het commando GRANT kunt u rechten voor tabellen en visies aan andere gebruikers toewijzen. U kunt de volgende rechten toewijzen:

- ALTER - Geeft de mogelijkheid om kolommen aan een tabel toe te voegen (kan niet voor een visie worden toegewezen).
- DELETE - Geeft de mogelijkheid om rijen uit een tabel of visie te verwijderen.
- INDEX - Geeft de mogelijkheid om het commando CREATE INDEX te gebruiken (kan niet voor een visie worden toegewezen).
- INSERT - Geeft de mogelijkheid om rijen in een tabel of visie in te voegen.
- SELECT - Geeft de mogelijheid om rijen uit een tabel of visie weer te geven.
- UPDATE - Geeft de mogelijkheid om een waarde uit een willekeurige of een specifieke kolom van een tabel of visie bij te werken.

U kunt rechten op de volgende manieren toewijzen:

- Door het sleutelwoord ALL te gebruiken (wat niet mogelijk is voor een visie), wijst u alle rechten in een keer toe.
- Door de <rechtenlijst> te gebruiken, waarin de toe te wijzen rechten gescheiden door komma's staan vermeld, kunt u deze afzonderlijk toewijzen.
- Door ON <tabellenlijst> te gebruiken, waarin de tabellen of visies waarvoor de rechten worden toegewezen tussen komma's staan vermeld, kunt u rechten voor één of meer tabellen of visies toewijzen.
- Door TO in combinatie met het sleutelwoord PUBLIC te gebruiken kunt u de rechten aan alle gebruikers toewijzen.
- Door TO in combinatie met <gebruikerslijst> te gebruiken, waarin de gebruikersnamen waaraan de rechten worden toegewezen tussen komma's staan vermeld, kunt u rechten aan afzonderlijke gebruikers toewijzen.

De sleutelwoorden PRIVILEGES en TABLE worden voor de duidelijkheid optioneel gebruikt.

OPMERKING
In een database die met een wachtwoord is beveiligd, berusten de rechten voor een tabel bij degene die deze heeft gemaakt en bij de gebruiker met de SQLDBA-gebruikersnaam. Beide gebruikers kunnen rechten voor de tabel met GRANT toewijzen en de tabel met DROP verwijderen. De gebruiker met de SQLDBA-gebruikersnaam beschikt over alle rechten voor alle tabellen, visies, synoniemen en indexen in iedere database.

Met GRANT toegewezen rechten zijn cumulatief. Dat wil zeggen dat u met GRANT rechten kunt toevoegen aan de rechten die een gebruiker al heeft. Aan een gebruiker kan meerdere malen hetzelfde recht worden toegewezen.

Door toevoeging van de clausule WITH GRANT OPTION geeft u degene aan wie u een recht toewijst, tegelijkertijd de mogelijkheid dit recht aan derden toe te wijzen. Wanneer u in een later stadium dit recht weer intrekt, wordt het recht bij alle derden aan wie het recht door de gemachtigde is toegekend, weer ingetrokken.

OPMERKING
Voor de clausule WITH GRANT OPTION geldt het volgende:

- *U kunt de clausule niet gebruiken in een commando waarmee rechten via PUBLIC aan alle gebruikers worden toegewezen.*
- *De clausule wordt genegeerd wanneer het UPDATE-recht is gespecificeerd met <kolommenlijst>.*

Voorbeelden

U kunt alle rechten voor de tabel Medew aan de gebruiker met de gebruikersnaam Janneke toewijzen door te typen:

```
SQL. GRANT ALL ON Medew
  TO Janneke;
```

U kunt het INSERT-, UPDATE-, en DELETE-recht voor de tabel Medew aan de gebruiker met de gebruikersnaam Gerard toewijzen door te typen:

```
SQL. GRANT INSERT, UPDATE, DELETE ON Medew
  TO Gerard;
```

Zie ook

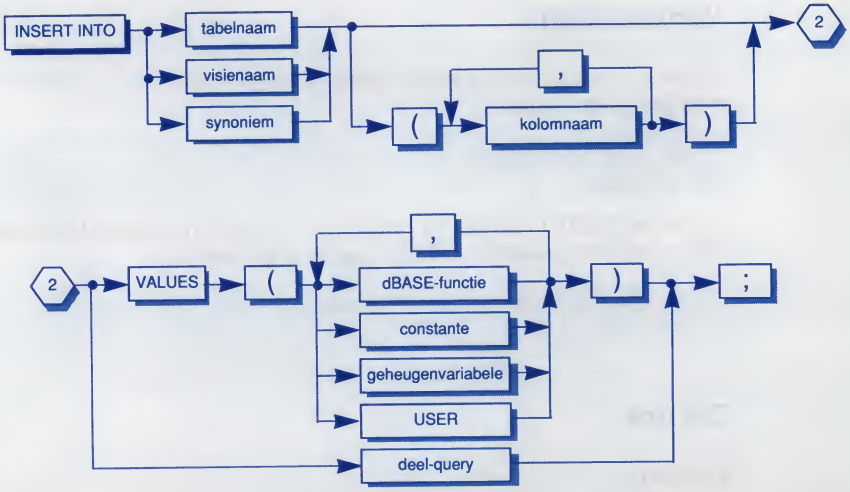
REVOKE

INSERT

Met het commando INSERT kunt u rijen in een tabel of bewerkbare visie invoegen. Het commando heeft twee vormen, die beide zowel interactief als in een ingebed SQL-programma kunnen worden uitgevoerd. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u over het INSERT-recht voor de opgegeven tabel of visie beschikken.

Syntaxis

```
INSERT INTO <tabelnaam>
    [(<kolommenlijst>)]
    VALUES (<waardenlijst>);                    (1)

INSERT INTO <tabelnaam>
    [(<kolommenlijst>)]
    <SELECT-commando>;                          (2)
```



Afbeelding 6-21 Het commando INSERT

Gebruik

U kunt met het commando INSERT rijen invoegen in een tabel of een bewerkbare visie (raadpleeg het gedeelte over het commando CREATE VIEW). U kunt zowel de tabel- of visienaam als een synoniem gebruiken.

In de eerste vorm van het commando gebruikt u <waardenlijst> om de afzonderlijke kolomwaarden op te geven die als nieuwe rij in de tabel moeten worden ingevoegd. In de tweede vorm voegt u kolomwaarden in voor één of meer rijen die u met het SELECT-commando uit een andere tabel hebt opgehaald.

Voor de waarden die in de waardenlijst worden opgegeven, geldt het volgende:

- De mogelijke waarden zijn constanten, dBASE-geheugenvariabelen, dBASE-functies die een waarde opleveren of het SQL-sleutelwoord USER (dat de gebruikersnaam van de actuele gebruiker oplevert).
- De waarden worden gescheiden door komma's. Tekenconstanten staan tussen enkele of dubbele aanhalingstekens (' of ") en datums staan tussen accoladen ({}) of vormen een onderdeel van de functie CTOD().
- De waarden moeten van het gegevenstype zijn dat voor de kolom waarin u invoegt, is gedefinieerd ("Hoogervorst" bijvoorbeeld moet worden ingevoegd in een kolom van het type CHAR en {15/04/90} in een kolom van het type DATE).

Wanneer u niet in iedere kolom van de tabel of visie een waarde invoegt, worden de niet opgegeven waarden als volgt geïnitieerd:

- kolom van het type CHAR; met een spatie;
- kolom van het type NUMERIC; met een 0;
- kolom van het type DATE; met een lege datumreeks (dd/mm/jj);
- kolom van het type LOGICAL; met de waarde onwaar (.F.).

U kunt met <kolommenlijst> de kolomnamen van de visie of tabel opgeven, waarin u rijen met waarden wilt invoegen. U hoeft de kolommenlijst niet op te geven als de volgorde van de waarden uit de waardenlijst of die waarin de kolomnamen bij het SELECT-commando zijn opgegeven, gelijk is aan de volgorde waarin de corresponderende kolommen van de tabel of visie zijn gedefinieerd.

Wanneer u niet voor iedere kolom van de tabel of visie een in te voegen waarde opgeeft (of met SELECT ophaalt), moet u wel een kolommenlijst specificeren. U hoeft een kolommenlijst niet op te geven in de volgorde waarin de kolommen in de tabel of visie zijn gedefinieerd. De kolommenlijst moet echter wel de volgorde hebben waarin de corresponderende kolomwaarden in de waardenlijst (of de SELECT-clausule) worden opgegeven.

Voorbeelden

Om één rij in de tabel Medew in te voegen typt u:

```
SQL. INSERT INTO Medew
     VALUES ("000013", "De Groot", "Albert", {15/07/90}, ;
     "Groningen", "000001", 5687, 3.5);
```

De melding **1 rij(en) ingevoegd** wordt weergegeven. Zoals u ziet, hoeft er geen kolommenlijst te worden opgegeven. De waarden worden immers ingevoegd in de volgorde waarin de corresponderende kolommen in de tabel Medew zijn gedefinieerd.

INSERT

U kunt een aantal rijen uit een bepaalde tabel in een andere tabel met dezelfde kolomindeling invoegen door te typen:

```
SQL. INSERT INTO Ad
     SELECT *
     FROM Medew
     WHERE Standpl = "Amsterdam";
```

Zie ook

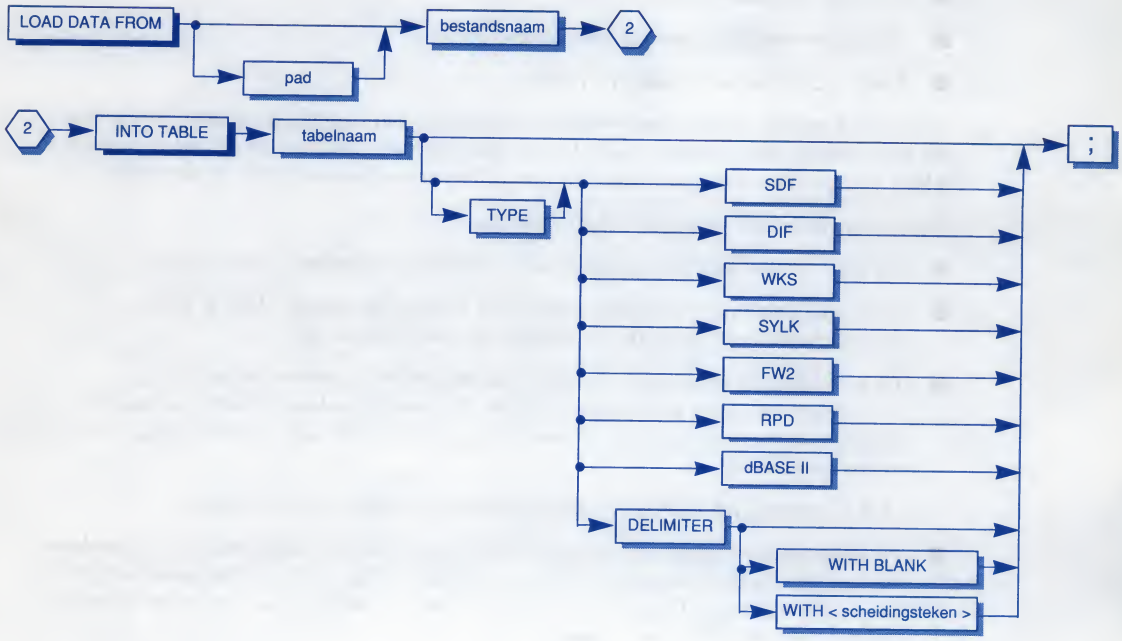
CREATE TABLE, CREATE VIEW, SELECT

LOAD DATA

Met het commando LOAD DATA kunt u gegevens uit een extern bestand importeren en aan een bestaande SQL-tabel in de actieve database toevoegen. Wanneer dBASE IV met een wachtwoord is beveiligd, dient u degene te zijn die de tabel heeft gemaakt of over het INSERT-recht voor de betreffende tabel te beschikken.

Syntaxis

LOAD DATA FROM [pad]<bestandsnaam>
INTO TABLE <tabelnaam>
[[TYPE]SDF/DIF/WKS/SYLK/FW2/RPD/DBASEII/
DELIMITED [WITH BLANK/WITH <scheidingsteken>]];



Afbeelding 6-22 Het commando LOAD DATA

LOAD DATA

Gebruik

Het commando LOAD DATA ondersteunt dezelfde bestandsstructuren als het dBASE IV-commando APPEND FROM. U kunt met LOAD DATA gegevens importeren uit de volgende soorten bestanden:

- dBASE II database-bestanden (dBASE II);
- dBASE III, dBASE III PLUS, en dBASE IV database-bestanden;
- RapidFile database-bestanden (RPD);
- Framework II database- en spreadsheet-bestanden (FW2);
- ASCII-bestanden, structuur met scheidingstekens (DELIMITED);
- ASCII-bestanden, structuur "System data" (SDF);
- VisiCalc structuurbestanden (DIF);
- Multiplan spreadsheet-structuurbestanden (SYLK);
- Lotus 1-2-3 structuurbestanden (WKS).

Het bestand waaruit de gegevens worden geïmporteerd, kan zowel in de actieve bestandsindex als in een andere bestandsindex staan. U moet deze dan met een expliciet pad opgegeven. Als u bij TYPE geen bestandstype opgeeft, wordt een database-bestand (.dbf) verondersteld.

Voor het invoerproces gelden de volgende regels:

- De SQL-tabel waarin de gegevens moeten worden geïmporteerd, moet bestaan.
- Het aantal kolommen in de tabel moet gelijk zijn aan het aantal velden in het te importeren bestand (tenzij dit een dBASE database-bestand is).
- De kolommen moeten wat de definities van breedte en gegevenstype betreft, overeenkomen met de te importeren velden. Wanneer de kolom te smal is, worden de gegevens uit het veld afgekapt. Wanneer de gegevenstypen niet overeenkomen, verschijnt er een foutmelding.

Voor het importeren van dBASE database-bestanden gelden de volgende regels:

- De kolomnamen van de tabel moeten overeenkomen met de veldnamen van het database-bestand. Het aantal kolommen in de tabel hoeft niet gelijk te zijn aan het aantal velden in het database-bestand. Er worden alleen gegevens geïmporteerd wanneer er een kolom en een veld met dezelfde naam bestaan.
- Memovelden worden niet geïmporteerd.
- Het te importeren bestand mag niet met dBASE IV noch met SQL zijn gecodeerd.
- Als het te importeren bestand een dBASE II database-bestand is, moet u het dBASE II-bestandstype opgeven.

U kunt gegevens uit spreadsheet-bestanden en bestanden met scheidingstekens alleen importeren als de structuur van de SQL-tabel overeenkomt met de structuur van het te importeren bestand. Spreadsheet-bestanden dienen te zijn opgeslagen in de *row-major* volgorde (het tegenovergestelde van de *column-major* volgorde).

Met de optie DELIMITED kan worden opgegeven dat de gegevens worden geïmporteerd uit een ASCII-tekstbestand (.txt) met scheidingstekens. Aan het te importeren bestand worden bij LOAD DATA (afhankelijk van hetgeen bij DELIMITED wordt opgegeven) de volgende eisen gesteld:

- DELIMITED - De bestandsvelden worden gescheiden door komma's en tekenvelden worden gescheiden door dubbele aanhalingstekens. Deze instelling kan ook worden opgegeven met DELIMITED WITH ".
- DELIMITED WITH BLANK - De bestandsvelden worden gescheiden door een spatie terwijl tekenvelden niet worden gescheiden.
- DELIMITED WITH <scheidingsteken> - De bestandsvelden worden gescheiden door komma's terwijl tekenvelden worden gescheiden door het opgegeven scheidingsteken. Het standaard scheidingsteken is het dubbele aanhalingsteken.

OPMERKING
Wanneer PROTECT is geïnstalleerd, geldt het volgende:

- *Gegevens worden door SQL automatisch gecodeerd zodra deze met LOAD DATA worden geïmporteerd (gegevens die met UNLOAD DATA naar bestanden worden geëxporteerd, worden niet gecodeerd).*
- *U kunt LOAD DATA niet gebruiken om gegevens te importeren uit een database-bestand dat is gecodeerd door SQL of dBASE IV (u kunt echter wel gegevens uit een niet-gecodeerde SQL-tabel importeren).*

Voorbeeld

Wanneer u gegevens uit een RapidFile-bestand aan de tabel Medew wilt toevoegen typt u:

```
SQL. LOAD DATA FROM \RAPID\RFDATA\Salmedew.rpd
     INTO TABLE Medew TYPE RPD;
```

Zie ook

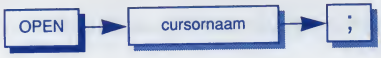
CREATE TABLE, UNLOAD DATA

OPEN

Met het commando OPEN wordt een met DECLARE gedefinieerde cursor geopend. Vervolgens wordt het bijbehorende SELECT-commando uitgevoerd, waarna de cursor in de eerste rij van de resultaattabel wordt geplaatst. Het commando kan alleen in ingebedde SQL-programma's worden uitgevoerd.

Syntaxis

OPEN <cursornaam>;



Afbeelding 6-23 Het commando OPEN

Gebruik

Wanneer dBASE IV met een wachtwoord is beveiligd, moet de gebruiker die het SQL-programma uitvoert, beschikken over de benodigde rechten voor de tabellen en visies die zijn vermeld in het SELECT-commando bij DECLARE CURSOR.

Een eenmaal geopende cursor blijft open en wordt pas onder de volgende omstandigheden gesloten:

- In het programma wordt het commando CLOSE <cursornaam> uitgevoerd.
- In het programma wordt de besturing weer overgedragen aan de SQL-aanwijzing of de commandostip (de cursor blijft dus geopend tijdens de uitvoering van subroutines van een lager niveau, ongeacht of het programmabestanden van dBASE (.prg) of van SQL (.prs) betreft).
- In het programma wordt het commando SET SQL OFF uitgevoerd.
- In het programma wordt één van de commando's CREATE DATABASE, START DATABASE of STOP DATABASE uitgevoerd.

Zodra het SELECT-commando is uitgevoerd, bevat de systeemvariabele Sqlcnt het aantal rijen in de resultaattabel. Wanneer de resultaattabel niet leeg is (Sqlcnt bevat een waarde groter dan 0), kan het commando FETCH worden gebruikt om de rijen uit de resultaattabel te verwerken.

OPMERKING
Een eenmaal gedefinieerde cursor kan onbeperkt worden geopend en gesloten. Voordat de cursor opnieuw wordt geopend, moet deze worden gesloten.

Voorbeeld

Nadat de cursor x_ptr met DECLARE is gedefinieerd, kunt u deze openen door te typen:

```
DECLARE x_ptr CURSOR
    FOR SELECT *
    FROM Medew
    WHERE Indienst > {01/09/86};
    .
    .
    .
    OPEN CURSOR x_ptr;
```

Zie ook

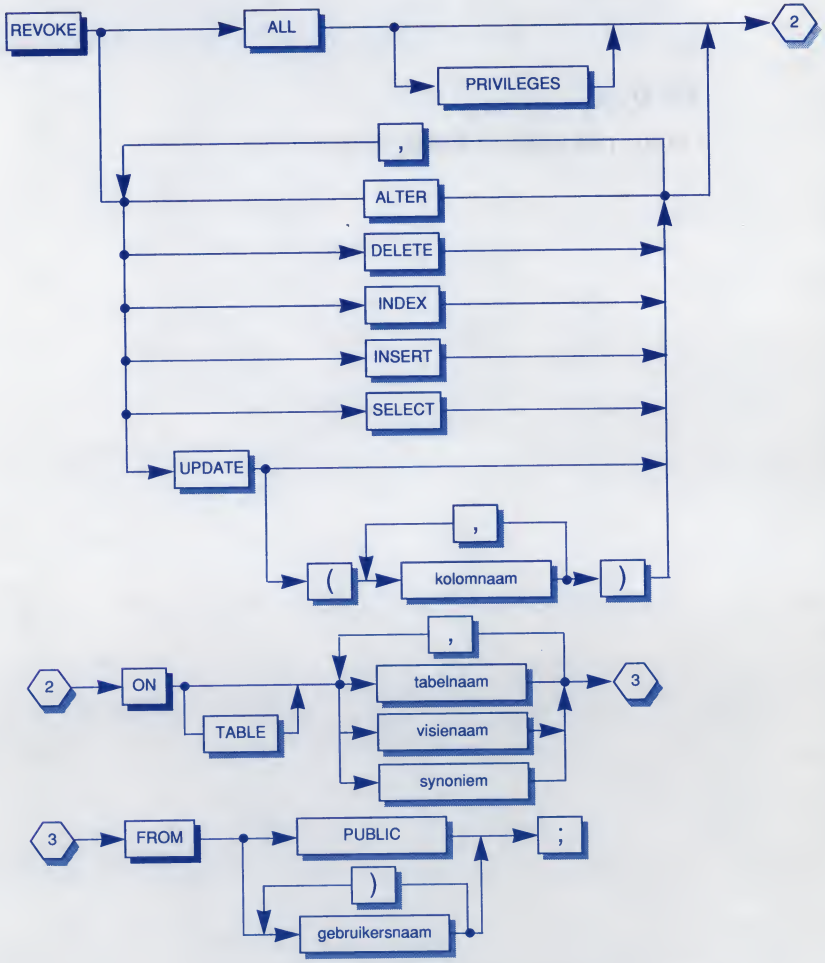
CLOSE, DECLARE CURSOR, FETCH

REVOKE

Met het commando REVOKE worden rechten voor tabellen en visies ingetrokken die in een eerder stadium met GRANT zijn toegewezen. De indeling van het commando REVOKE is vergelijkbaar met die van het commando GRANT. In de rechtenlijst kunnen dezelfde sleutelwoorden staan als die welke bij GRANT zijn toegestaan.

Syntaxis

REVOKE ALL [PRIVILEGES]/<rechtenlijst>
    ON [TABLE]<tabellenlijst>
    FROM PUBLIC/<gebruikerslijst>;



Afbeelding 6-24 Het commando REVOKE

Gebruik

U kunt rechten met REVOKE op de volgende manieren intrekken:

- Door het sleutelwoord ALL te gebruiken trekt u alle rechten in.
- Door <rechtenlijst> te gebruiken, waarin de in te trekken rechten gescheiden door komma's staan vermeld, kunt u deze afzonderlijk intrekken.
- Door ON <tabellenlijst> te gebruiken, waarin de tabellen of visies waarvoor de rechten worden ingetrokken tussen komma's staan vermeld, kunt u rechten voor één of meer tabellen of visies intrekken.
- Door FROM in combinatie met het sleutelwoord PUBLIC te gebruiken kunt de rechten van alle gebruikers intrekken.
- Door FROM in combinatie met <gebruikerslijst> te gebruiken, waarin de gebruikersnamen van de gebruikers wier rechten worden ingetrokken tussen komma's staan vermeld, kunt u rechten van afzonderlijke gebruikers intrekken.

De sleutelwoorden PRIVILEGES en TABLE worden voor de duidelijkheid optioneel gebruikt.

U kunt alleen rechten intrekken die u in een eerder stadium hebt toegewezen (zie de opmerking). Een gebruiker aan wie een bepaald recht met de optie WITH GRANT OPTION is toegewezen, kan slechts de rechten intrekken die hij met deze optie aan derden heeft toegewezen.

OPMERKING
Een gebruiker kan de rechten van degene die een tabel of visie heeft gemaakt of van de SQLDBA-gebruikersnaam, niet intrekken. De gebruiker met de SQLDBA-gebruikersnaam kan elk recht intrekken dat aan een gebruiker is toegewezen, ook die rechten die hij niet zelf heeft toegewezen.

Wanneer u rechten intrekt die aan een gebruiker zijn toegewezen met de optie WITH GRANT OPTION, worden ook de rechten ingetrokken die deze gebruiker aan derden heeft toegewezen. Wanneer u aan een gebruiker bepaalde rechten met de optie WITH GRANT OPTION hebt toegewezen en u slechts die optie wilt intrekken, moet u eerst met de optie ALL alle rechten intrekken. Vervolgens wijst u met GRANT opnieuw deze rechten toe, maar zonder de optie WITH GRANT OPTION.

Als een recht door meer gebruikers is toegewezen, moet het door alle gebruikers worden ingetrokken. Het commando REVOKE ..FROM PUBLIC trekt alleen de rechten in die aan alle gebruikers zijn toegewezen. Afzonderlijk toegewezen rechten blijven van kracht.

Voorbeelden

U kunt de rechten voor de tabel Medew, die met GRANT via de optie PUBLIC aan alle gebruikers zijn toegewezen, weer intrekken. U typt dan:

```
SQL. REVOKE ALL ON Medew
     FROM PUBLIC;
```

U kunt de INSERT- en UPDATE-rechten voor de tabel Medew die aan de gebruiker met de gebruikersnaam Jan zijn toegewezen, weer intrekken. U typt dan:

```
SQL. REVOKE INSERT, UPDATE ON Medew
     FROM Jan;
```

Zie ook

GRANT

ROLLBACK

Als een transactie is afgebroken, kunt u tabellen met het commando ROLLBACK weer in de oorspronkelijke staat terugbrengen

Syntaxis

ROLLBACK [WORK];

Afbeelding 6-25 Het commando ROLLBACK

Gebruik

U kunt met het commando ROLLBACK wijzigingen ongedaan maken die in het kader van een transactie (BEGIN/END TRANSACTION) door SQL-commando's zijn aangebracht. Het sleutelwoord WORK kan voor de duidelijkheid worden opgenomen.

OPMERKING
U kunt alle SQL-commando's opnemen in een transactie met uitzondering van de volgende commando's:

- *commando's voor het definiëren van gegevens (raadpleeg het gedeelte over commandocategorieën eerder in dit hoofdstuk);*
- *commando's voor het starten van databases (START DATABASE, STOP DATABASE);*
- *hulpprogramma's (raadpleeg het gedeelte over commandocategorieën eerder in dit hoofdstuk);*
- *commando's voor de beveiliging van databases (GRANT en REVOKE).*

Voorbeelden

Bekijk het volgende stuk SQL-programmacode:

```
ON ERROR DO Herstel
SET REPROCESS TO 15
BEGIN TRANSACTION
    UPDATE Medew
    SET Commissie = Commissie * 1.5
    WHERE Prov = "Drenthe";
END TRANSACTION
ON ERROR
IF COMPLETED()
    @ 21,15 SAY "Transactie succesvol beëindigd"
ENDIF
```

Stel dat er tijdens de uitvoering van het commando UPDATE in de transactie fouten zouden optreden en dat u de tabel Medew wilt terugbrengen in de oorspronkelijke staat. U moet dan in de foutherstelprocedure Herstel het commando ROLLBACK opnemen.

```
PROCEDURE Herstel
    @21,15 SAY "Er is een fout opgetreden in uw transactie"
    @22,15 SAY "Wilt u het opnieuw proberen? (J/N)" GET keuze PICTURE "!"
    READ
    @21,15 TO 22,65 CLEAR
    IF keuze = "J"
        RETRY
    ELSE
        @21,15 SAY "Even geduld. Uw transactie wordt hersteld"
        ROLLBACK;
    ENDIF
RETURN
```

Met het commando RUNSTATS kunt u de statistieken van één of meer tabellen en de bijbehorende indexen uit de actieve database bijwerken. Deze statistieken zijn opgeslagen in de SQL-systeemcatalogustabellen.

Syntaxis

RUNSTATS [<tabelnaam>];

Afbeelding 6-26 Het commando RUNSTATS

Gebruik

Statistieken worden bij het maken van tabellen en indexen opgeslagen in de SQL-catalogustabellen. Met behulp van de statistieken bepaalt SQL de meest efficiënte manier om database-bewerkingen uit te voeren. Bij wijziging van de database worden de statistieken niet automatisch gewijzigd. U moet om de statistieken bij te werken regelmatig het commando RUNSTATS uitvoeren.

Met het commando RUNSTATS kunt u ook beschadigde indexen opsporen en ingaven in de systeemcatalogus vergelijken met indexlabels in samengestelde indexbestanden. Wanneer er onregelmatigheden worden vastgesteld, wordt er een melding weergegeven.

U moet het commando RUNSTATS met <tabelnaam> uitvoeren nadat u een van de volgende wijzigingen in een tabel hebt aangebracht:

- U hebt met het commando ALTER TABLE de tabeldefinities gewijzigd.
- U hebt met de commando's INSERT, DELETE of UPDATE in meer dan tien procent van de rijen gegevens gewijzigd.
- U hebt met het commando CREATE INDEX nieuwe indexen gemaakt.

Wanneer u het commando RUNSTATS uitvoert zonder een tabelnaam op te geven, worden de statistieken van alle tabellen en indexen in de actieve database bijgewerkt. Wanneer u RUNSTATS uitvoert terwijl het database-bestand van een van de tabellen in de actieve database in de dBASE-modus is geopend, wordt er een melding weergegeven waarin staat dat het bestand al is geopend.

OPMERKING
Raadpleeg hoofdstuk 7 voor meer informatie over de SQL-catalogustabellen en over de statistieken die met RUNSTATS worden bijgewerkt.

Voorbeelden

U kunt de statistieken van alle tabellen in de actieve database bijwerken door te typen:

```
SQL. RUNSTATS;
```

U kunt de statistieken van de tabel Medew bijwerken door te typen:

```
SQL. RUNSTATS Medew;
```

SELECT

Met het commando SELECT kunt u gegevens ophalen. Deze kunnen in een resultaattabel worden weergegeven of in andere SELECT-zoekvoorwaarden of SQL-programmavariabelen worden gebruikt. U kunt SELECT-commando's interactief of in ingebedde SQL-programma's uitvoeren. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u degene zijn die de tabellen of visies waarnaar bij het SELECT-commando wordt verwezen, heeft gemaakt, of het SELECT-recht voor genoemde visies of tabellen hebben.

Syntaxis

```
SELECT <clausule>
    [INTO <clausule>]
    [FROM <clausule>]
    [WHERE <clausule>]
    [GROUP BY <clausule>]
    [HAVING <clausule>]
    [UNION <SELECT-commando>...]
    [ORDER BY <clausule>/FOR UPDATE OF <clausule>]
    [SAVE TO TEMP <clausule>];
```

In de volgende gedeelten worden de clausules van het SELECT-commando afzonderlijk behandeld.

De clausule SELECT

SELECT is een verplichte clausule waarin de kolommen, de SQL-verzamelfuncties en de uitdrukkingen worden opgegeven die in de resultaattabel van het SELECT-commando moeten worden opgenomen. De uitgebreide syntaxis voor de clausule SELECT is als volgt:

```
SELECT [ALL/DISTINCT] <kolommenlijst>/*
```

Kolommen waarvan de naam in de kolommenlijst is opgenomen, moeten bestaan in de tabellen of visies die worden opgenomen in de clausule FROM. Kolomnamen moeten door komma's worden gescheiden.

In de kolommenlijst kunnen kolomnamen meerdere keren voorkomen. In dat geval komen deze in verschillende bij FROM opgegeven tabellen en visies voor. Het is dan nodig de tabelnaam bij de kolomnaam op te geven.

Zo wordt de kolom Medew_nr uit de tabel Medew opgegeven als Medew.Medew_nr, terwijl de kolom Medew_nr uit de tabel Verkopen als Verkopen.Medew_nr wordt opgegeven. Om niet de complete namen van tabellen aan de kolomnamen in de kolommenlijst te hoeven toevoegen kunt u alias-namen en synoniemen voor de tabellen en visies in de clausule FROM opgeven.

Als u een asterisk (*) gebruikt in plaats van de kolommenlijst, geeft u op dat alle kolommen uit de tabellen en visies van de clausule FROM in het resultaat moeten worden opgenomen. U kunt aan de asterisk ook de naam toevoegen van een tabel waarvan u alle kolommen wilt ophalen (bijvoorbeeld Klanten.*).

Het optionele sleutelwoord ALL geeft aan dat alle rijen moeten worden weergegeven. Wanneer u het SELECT-commando gebruikt zonder met de clausule WHERE het aantal rijen te beperken, worden volgens de standaardinstelling alle rijen weergegeven.

Wanneer in een tabel identieke rijen voorkomen, kunt u het optionele sleutelwoord DISTINCT opgeven. Daarmee geeft u aan dat de rij maar eenmaal hoeft te worden weergegeven. Als u DISTINCT gebruikt, mag de totale aaneengeschakelde breedte van de kolommenlijst niet meer dan 100 bytes in beslag nemen. Voor het gebruik van DISTINCT gelden de volgende regels:

- DISTINCT kan in de clausule SELECT van een SELECT-commando maar eenmaal worden gebruikt. Dit geldt voor hogere en lagere queries.
- DISTINCT is alleen van toepassing op de eerste kolom die in een clausule SELECT is opgegeven.

Naast kolomnamen en de asterisk kan de clausule SELECT uitdrukkingen en verzamelfuncties bevatten. Als uitdrukking zijn de volgende elementen geldig:

- een rekenkundige formule waarmee waarden in een rekenkolom worden opgegeven (bijvoorbeeld *Salaris*12)*;
- een dBASE-functie die kolomwaarden bewerkt (bijvoorbeeld *UPPER(Stad))*;
- een constante (bijvoorbeeld *"(Jaarsalaris)")* waarmee de gegevens in een aangrenzende kolom worden omschreven;
- een dBASE-geheugenvariabele;
- het sleutelwoord USER, dat de gebruikersnaam van de huidige gebruiker oplevert als dBASE IV met een wachtwoord is beveiligd.

Een verzamelfunctie bestaat uit één van de sleutelwoorden AVG, COUNT, MAX, MIN en SUM gevolgd door een tussen haken geplaatste kolomnaam (bijvoorbeeld *MAX(salaris)*. Wanneer de clausule SELECT afzonderlijke kolomnamen en verzamelfuncties bevat, moet bij het SELECT-commando verplicht de clausule GROUP BY worden gebruikt.

Voor elke kolom in de resultaattabel wordt een kop weergegeven. De volgende elementen kunnen als kop worden weergegeven:

- Voor een kolom die u in de clausule SELECT opgeeft, wordt de kolomnaam weergegeven, al dan niet uitgebreid met de tabelnaam.
- Voor een uitdrukking die u in de clausule SELECT opgeeft, wordt het prefix *EXP* weergegeven, voorzien van het positienummer van de uitdrukking in de clausule (geteld van links naar rechts; bijvoorbeeld *EXP1*, *EXP2* enzovoort).
- Voor een verzamelfunctie die u in de clausule SELECT opgeeft, wordt de functienaam weergegeven, voorzien van het positienummer van de functie in de clausule (geteld van links naar rechts; bijvoorbeeld *MAX1*, *MIN2*, *AVG3* enzovoort).
- Voor een kolom die u in de clausule GROUP BY opgeeft, wordt het prefix *G_ weergegeven*, gevolgd door de naam van de kolom waarvan de waarden worden gebruikt voor het groeperen van de resultaattabel (bijvoorbeeld *G_Beschr*).

OPMERKING

1. *U kunt voor tabelnamen synoniemen en alias-namen opgeven. Op die manier kunt u meer kolomnamen in de clausule SELECT opnemen, met name dan wanneer de kolomnamen moeten worden voorzien van de tabelnaam.*
2. *U moet SET PAUSE op ON instellen wanneer u brede resultaattabellen wilt weergeven. Op die manier worden de gegevens scherm voor scherm weergegeven. U kunt het verschuiven ook met* **Ctrl-S** *stoppen en voortzetten.*

De clausule INTO

De clausule INTO wordt alleen in ingebedde SQL-programma's gebruikt om SELECT-resultaten per rij te verwerken. De uitgebreide syntaxis voor de clausule INTO is als volgt:

INTO <geheugenvariabele1>[,<geheugenvariabele2>,...]

Raadpleeg het gedeelte over ingebedde SELECT voor een enkele rij verderop in dit gedeelte voor meer informatie over de werking van INTO.

De clausule FROM

FROM is een verplichte clausule waarmee de tabellen en visies worden opgegeven die de gegevens voor de resultaattabel bevatten. De uitgebreide syntaxis voor de clausule FROM is als volgt:

FROM <tabelnaam>/<visienaam> [alias][,<tabelnaam>/<visienaam> [alias]...]

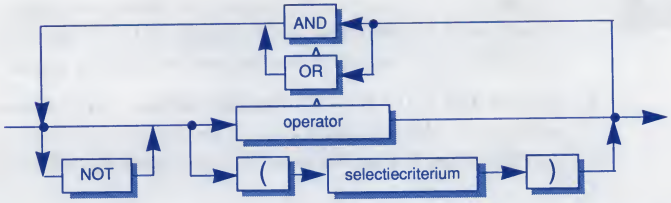
De clausule FROM is ook optioneel te gebruiken om aliasnamen voor de tabellen en visies in de clausule te definiëren. Een aliasnaam is een alternatieve naam die in de plaats van tabelnamen en visienamen in een gecorreleerde sub-query of "self-join" wordt gebruikt. Een alias-naam kan uit maximaal tien tekens bestaan, kan uit letters, cijfers en onderstrepingstekens bestaan en moet met een letter beginnen.

OPMERKING
U kunt als aliasnaam geen enkele letter gebruiken die tussen A en J ligt.

De clausule WHERE

Met de optionele clausule WHERE wordt een selectiecriterium opgegeven waaraan rijen moeten voldoen om in de resultaattabel te worden opgenomen. De uitgebreide syntaxis voor de clausule WHERE is als volgt:

WHERE [NOT]<selectiecriterium>
    [AND/OR [NOT]<selectiecriterium>...]



Afbeelding 6-27 De clausule WHERE

Om in aanmerking te komen voor opname in de resultaattabel moet een rij in een tabel of visie voldoen aan het met WHERE opgegeven selectiecriterium. Dat wil zeggen dat het selectiecriterium voor een bepaalde rij waar moet zijn. Als het criterium onwaar is, wordt de rij niet opgenomen.

Een selectiecriterium kan elke geldige uitdrukking zijn en kan bestaan uit kolomnamen, dBASE-functies (*geen* verzamelfuncties), dBASE-geheugenvariabelen, operatoren, haakjes en constanten. Zo is *Achtnaam = "Willemse"* een enkelvoudig selectiecriterium, waarmee alleen rijen worden geselecteerd waarvan "Willemse" de waarde van de kolom Achtnaam is. In afbeelding 6-28 ziet u een overzicht van de SQL-uitdrukkingen.

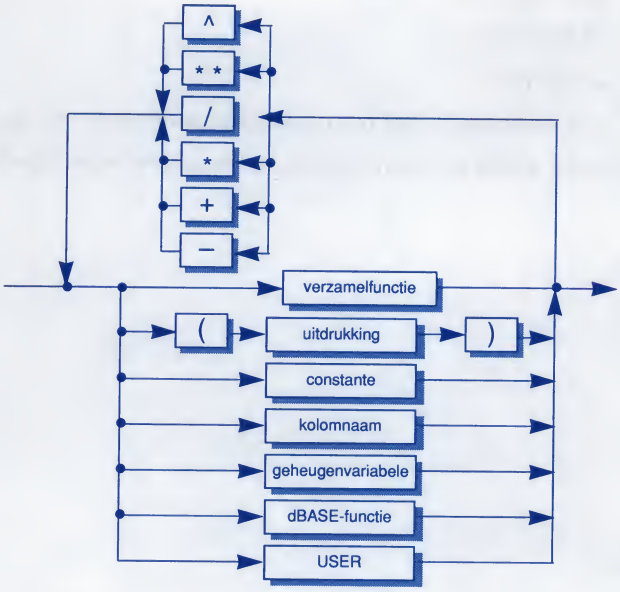
U kunt in de clausule WHERE een SELECT-commando gebruiken dat de waarden voor het selectiecriterium moet opleveren (bijvoorbeeld *WHERE Salaris > (SELECT AVG(Salaris) from Medew*).

U kunt de volgende vergelijkingsoperatoren in een selectiecriterium gebruiken:

Operator	Omschrijving
<	kleiner dan
>	groter dan
<=	kleiner dan of gelijk aan
>=	groter dan of gelijk aan
=	gelijk aan
<>, # of !=	niet gelijk aan
!>	niet groter dan
!<	niet kleiner dan

U kunt alleen vergelijkingen maken tussen uitdrukkingen met compatibele gegevenstypen. Een numerieke kolomwaarde bijvoorbeeld moet met een numerieke waarde worden vergeleken, en een kolomwaarde van het type CHAR met een tekenwaarde of een tekenreeks. Met de dBASE-functies VAL() en STR() kunt u ook niet-compatibele gegevenstypen met elkaar vergelijken.

Wanneer u in selectiecriteria tekenreeksen opgeeft, moeten deze tussen enkele of dubbele aanhalingstekens staan.



Afbeelding 6-28 SQL-uitdrukkingen

U kunt selectiecriteria combineren met de logische operatoren AND, OR en NOT. Wanneer u het selectiecriterium *Achtnaam<>"Willemse" AND Leeftijd < 34* opgeeft, moet een rij aan beide criteria voldoen om in de resultaattabel te worden opgenomen.

De volgorde waarin gecombineerde selectiecriteria worden geëvalueerd, wordt bepaald door de prioriteit van de operatoren waarmee de criteria worden verbonden. Het NOT-criterium wordt als eerste geëvalueerd, gevolgd door het AND-criterium, op zijn beurt gevolgd door het OR-criterium.

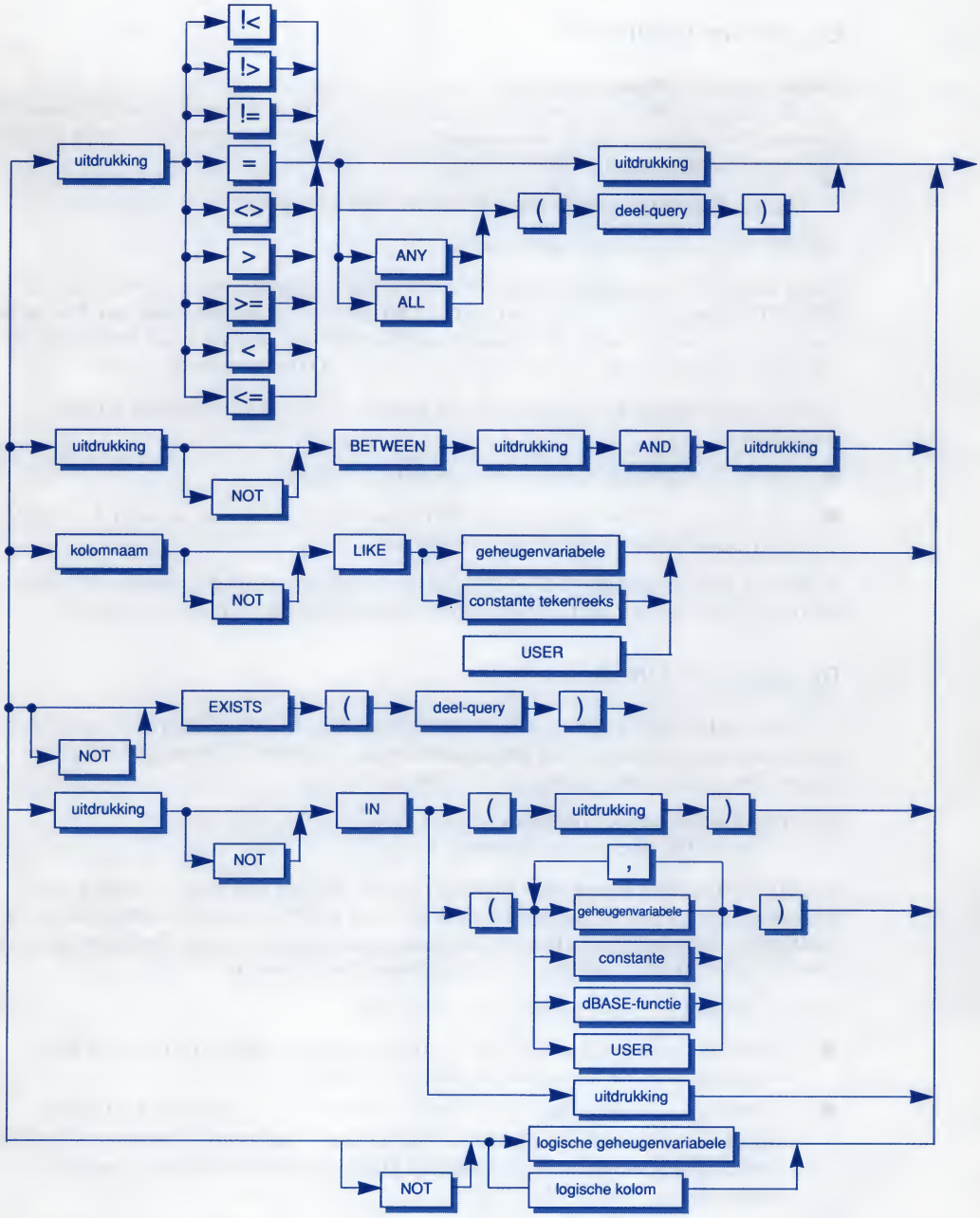
U kunt criteria met haakjes groeperen en zo de volgorde waarin deze worden geëvalueerd, wijzigen.

OPMERKING
Bij waarden van het type CHAR worden de ASCII-waarden van de tekens van links naar rechts geëvalueerd. Doordat de ASCII-waarden van hoofdletters lager zijn dan die van kleine letters wordt de Z als lager geëvalueerd dan de a. Ook geldt dat Z niet gelijk is aan z.

In de selectiecriteria die u met WHERE opgeeft, kunnen ook bijvoorbeeld ALL, ANY, BETWEEN, EXISTS, IN en LIKE worden opgenomen. Deze operatoren kunnen in combinatie met de volgende elementen worden gebruikt:

- dBASE-geheugenvariabelen;
- dBASE-functies;
- sub-queries;
- het sleutelwoord USER (dat de gebruikersnaam van de huidige gebruiker oplevert).

In afbeelding 6-29 ziet u een overzicht van de gebruiksmogelijkheden van opratoren.



Afbeelding 6-29 SQL-operatoren

De clausule GROUP BY

Met de optionele clausule GROUP BY kunt u de gegevens van een aantal rijen waarin de waarde van een bepaalde kolom gelijk is, samenvatten. Elke groep wordt als één rij in de resultaattabel weergegeven. De kolomwaarden van die rij zijn karakteristiek voor de groep of zijn een verzamelde waarde van de groep.

De uitgebreide syntaxis voor de clausule GROUP BY is als volgt:

GROUP BY <kolomnaam>[,<kolomnaam>...]

Iedere kolomnaam die u bij de parameter <kolomnaam> opgeeft, moet ook in de clausule SELECT zijn opgegeven. U kunt met GROUP BY geen kolom opgeven waarvan de waarden met een rekenkundige functie of verzamelfunctie worden afgeleid. De totale breedte van de bij GROUP BY opgegeven kolommen mag niet meer dan 100 bytes in beslag nemen.

In de volgende SELECT-commando's is het gebruik van de clausule GROUP BY niet toegestaan:

- SELECT-commando's die een clausule INTO bevatten;
- een sub-query waarvan de clausule FROM verwijst naar een visie waarvan de definitie een clausule GROUP BY of HAVING bevat.

In een SELECT-commando (ook wanneer dat in een hogere query en in sub-queries wordt gebruikt) kunt u per SELECT-clausule slechts een clausule GROUP BY opgeven.

De clausule HAVING

De clausule HAVING is een optionele clausule waarmee een selectiecriterium wordt opgegeven om de rijen die in een gegroepeerd resultaat verschijnen, te beperken. De uitgebreide syntaxis voor de clausule HAVING is als volgt:

HAVING [NOT]<selectiecriterium>
    [AND/OR [NOT]<selectiecriterium>...]

Met HAVING geeft u één of meer selectiecriteria op die voor elke rij in de groep waar of onwaar zijn. Als er met de clausule GROUP BY geen groep is vastgelegd, wordt het gehele resultaat als een enkele groep beschouwd. In dat geval wordt doorgaans elke kolom die in de clausule SELECT is opgegeven, door een verzamelfunctie bewerkt.

Voor de clausule HAVING gelden de volgende regels:

- Doorgaans correspondeert elk bij HAVING opgegeven selectiecriterium met een verzamelfunctie die bij de clausule SELECT is opgegeven.
- In het selectiecriterium dat bij HAVING wordt opgegeven, is een SELECT-query toegestaan. Bij de clausule FROM van die sub-query mag echter geen tabel- of visienaam worden opgegeven die al bij de clausule FROM van de hogere SELECT-query is opgegeven.
- Een gecorreleerde sub-query kan niet in een met HAVING opgegeven selectiecriterium worden gebruikt.

De clausule UNION

Met de optionele clausule UNION kunt u de resultaattabellen van twee of meer SELECT-commando's samenvoegen. U maakt zo één enkele resultaattabel zonder dubbele rijen. De complete syntaxis voor de query met een clausule UNION is als volgt:

```
<SELECT-commando> [UNION <SELECT-commando>...]
    [ORDER BY <clausule>]
    [SAVE TO TEMP <clausule> KEEP];
```

Bij het koppelen van SELECT-commando's met de clausule UNION gelden de volgende regels:

- Elke kolom die in de clausule SELECT van een SELECT-commando is opgegeven, moet corresponderen met een kolom die in de clausule SELECT van een ander SELECT-commando is opgegeven. De kolommen hoeven evenwel niet dezelfde naam te hebbben. De clausule UNION kan niet worden gebruikt voor een kolom van het gegevenstype LOGICAL.
- Corresponderende kolommen moeten overeenkomen in gegevenstype en breedte. Kolommen van het gegevenstype NUMERIC moeten beide hetzelfde aantal cijfers bevatten. Decimale getallen moeten beide hetzelfde aantal decimalen hebben.
- In een clausule UNION zijn dBASE-geheugenvariabelen en dBASE-functies niet toegestaan

Het resultaat van een UNION-bewerking wordt gesorteerd op basis van de waarden van alle kolommen die bij de betrokken SELECT-commando's zijn opgegeven. U kunt een andere sorteervolgorde opgeven door na het laatste SELECT-commando de clausule ORDER BY op te geven. U moet geen kolomnaam opgegeven, maar het cijfer dat de positie van de kolom in de clausule SELECT (geteld van links naar rechts) aangeeft (bijvoorbeeld *ORDER BY 1*) (raadpleeg het gedeelte over de clausule ORDER BY).

Met de clausule SAVE TO TEMP kunt u de resultaten van UNION voor gebruik tijdens de actieve sessie in een tijdelijke tabel opslaan. Met de optie KEEP kunt u het resultaat als een dBASE-bestand permanent opslaan (raadpleeg het gedeelte over de clausule SAVE TO TEMP).

De clausule ORDER BY

De clausule ORDER BY biedt u de mogelijkheid om de volgorde van de rijen in een resultaattabel op te geven. U kunt als volgorde oplopend (ASC, standaardinstelling), afdalend (DESC), of een combinatie van beide opgeven.

De uitgebreide syntaxis voor de clausule ORDER BY is als volgt:

```
[ORDER BY <kolomnaam>/<geheel getal> [ASC/DESC]
    [,<kolomnaam>/<geheel getal> [ASC/DESC]...];
```

Een kolom die bij een clausule ORDER BY wordt opgegeven, moet ook bij de clausule SELECT zijn opgegeven. De totale lengte van de bij ORDER BY opgegeven kolommen mag niet meer dan 100 bytes in beslag nemen.

De kolom die u bij ORDER BY opgeeft, kunt u niet alleen specificeren met een kolomnaam, maar ook met het cijfer dat de positie (geteld van links naar rechts) van die kolom in de clausule SELECT aangeeft. Het opgeven van een cijfer is nodig om resultaten te kunnen sorteren op basis van berekende kolomwaarden, waarden die zijn afgeleid door gebruik van constanten, dBASE-functies, of geheugenvariabelen, en op basis van andere uitdrukkingen.

Als u bij de clausule ORDER BY meer dan één kolom opgeeft, worden de rijen van de resultaattabel eerst gerangschikt op basis van de waarden in de als eerste vermelde kolom. Rijen met dezelfde waarde in de eerste kolom worden gerangschikt op grond van de waarden in de als tweede vermelde kolom enzovoort. U kunt voor iedere kolom die u bij ORDER BY opgeeft, oplopende of afdalende sorteervolgorde specificeren.

De clausule FOR UPDATE OF

De clausule FOR UPDATE OF is een optionele clausule die in het SELECT-commando van het commando DECLARE CURSOR wordt gebruikt. De clausule kan dus alleen in ingebedde SQL-programma's worden gebruikt.

Met FOR UPDATE OF worden kolommen opgegeven die kunnen worden bijgewerkt met behulp van een navolgend commando UPDATE waarin de clausule WHERE CURRENT OF is opgenomen. De volledige syntaxis voor de clausule FOR UPDATE OF is als volgt:

FOR UPDATE OF <kolomnaam> [,<kolomnaam>...]

De bij FOR UPDATE OF vermelde kolommen moeten deel uitmaken van de tabel of bewerkbare visie die bij de clausule FROM van het SELECT-commando is opgegeven. In SELECT-commando's die de clausule INTO, ORDER BY of SAVE TO TEMP bevatten, is de clausule FOR UPDATE OF niet toegestaan.

De clausule SAVE TO TEMP

Met de optionele clausule SAVE TO TEMP kunt u de resultaten van SELECT opslaan. U kunt opslaan in een tijdelijke tabel die alleen gedurende de actieve sessie bruikbaar is, maar ook in in een permanent dBASE-bestand. De uitgebreide syntaxis voor de clausule SAVET TO TEMP is als volgt:

SAVE TO TEMP <tabelnaam> [(<kolomnaam>[,<kolomnaam>...])]
[KEEP]

Met SAVE TO TEMP wordt het resultaat opgeslagen in een tabel met de naam die bij <tabelnaam> is opgegeven. Voor het benoemen van tijdelijke tabellen gelden dezelfde regels als voor permanente tabellen (raadpleeg het gedeelte over CREATE TABLE). De gegevenstypen en de breedte van de kolommen zijn gelijk aan die welke voor de kolommen van de basistabel of visie zijn gedefinieerd.

U kunt de optionele specificatie <kolomnaam> op een aantal manieren gebruiken:

- U kunt namen toewijzen aan kolommen waarvan de waarden worden afgeleid door gebruik van een uitdrukking, dBASE-functie of verzamelfunctie, geheugenvariabele of constante (voor deze typen kolommen moet u <kolomnaam> opgeven).

- U kunt namen van kolommen die in de basistabel of visie zijn gedefinieerd, wijzigen.

Door de optie KEEP weg te laten heeft de resultaattabel de status tijdelijk. U kunt zo'n tijdelijk tabel op de volgende manieren gebruiken:

- In de interactieve SQL-modus is de tabel te gebruiken zolang u dezelfde database in de actieve SQL-sessie gebruikt. Zodra u van database verandert of terugkeert naar de dBASE-commandostip, wordt de tabel verwijderd.
- In de ingebedde SQL-modus is de tabel te gebruiken totdat het programma de besturing weer overdraagt aan de commandostip of de SQL-aanwijzing.

De optie KEEP heeft tot gevolg dat de tijdelijke resultaattabel als permanent niet-gecodeerd dBASE-database-bestand in de actieve bestandsindex wordt opgeslagen. U kunt dit bestand in dBASE IV gebruiken (bijvoorbeeld voor een rapport), maar u kunt er met DBDEFINE ook een nieuwe SQL-tabel van maken.

OPMERKING
Bij een SELECT-commando dat een clausule FOR UPDATE OF of INTO bevat, is de clausule SAVE TO TEMP niet toegestaan.

Met de clausule SAVE TO TEMP wordt, wanneer SET CATALOG op ON is ingesteld, een ingave voor de tijdelijke tabel aan het geopende dBASE-catalogusbestand toegevoegd. Wanneer de tabel wordt verwijderd, wordt ook de ingave verwijderd, tenzij de KEEP optie is gebruikt. In dat geval wordt de ingave voor de tijdelijke tabel vervangen door een ingave voor het nieuwe database-bestand.

Gebruik

SELECT is het SQL-commando met het breedste toepassingsgebied. Het kan als zelfstandig commando worden gegeven, maar ook als sub-query deel uitmaken van commando's als INSERT, UPDATE en DELETE.

De syntaxis voor het SELECT-commando varieert, afhankelijk van de manier waarop het wordt gebruikt. Mogelijke gebruikswijzen zijn:

- SELECT voor alleen weergave;
- ingebed SELECT voor een enkele rij;
- ingebed SELECT gebruikt met DECLARE CURSOR.

SELECT voor alleen weergave

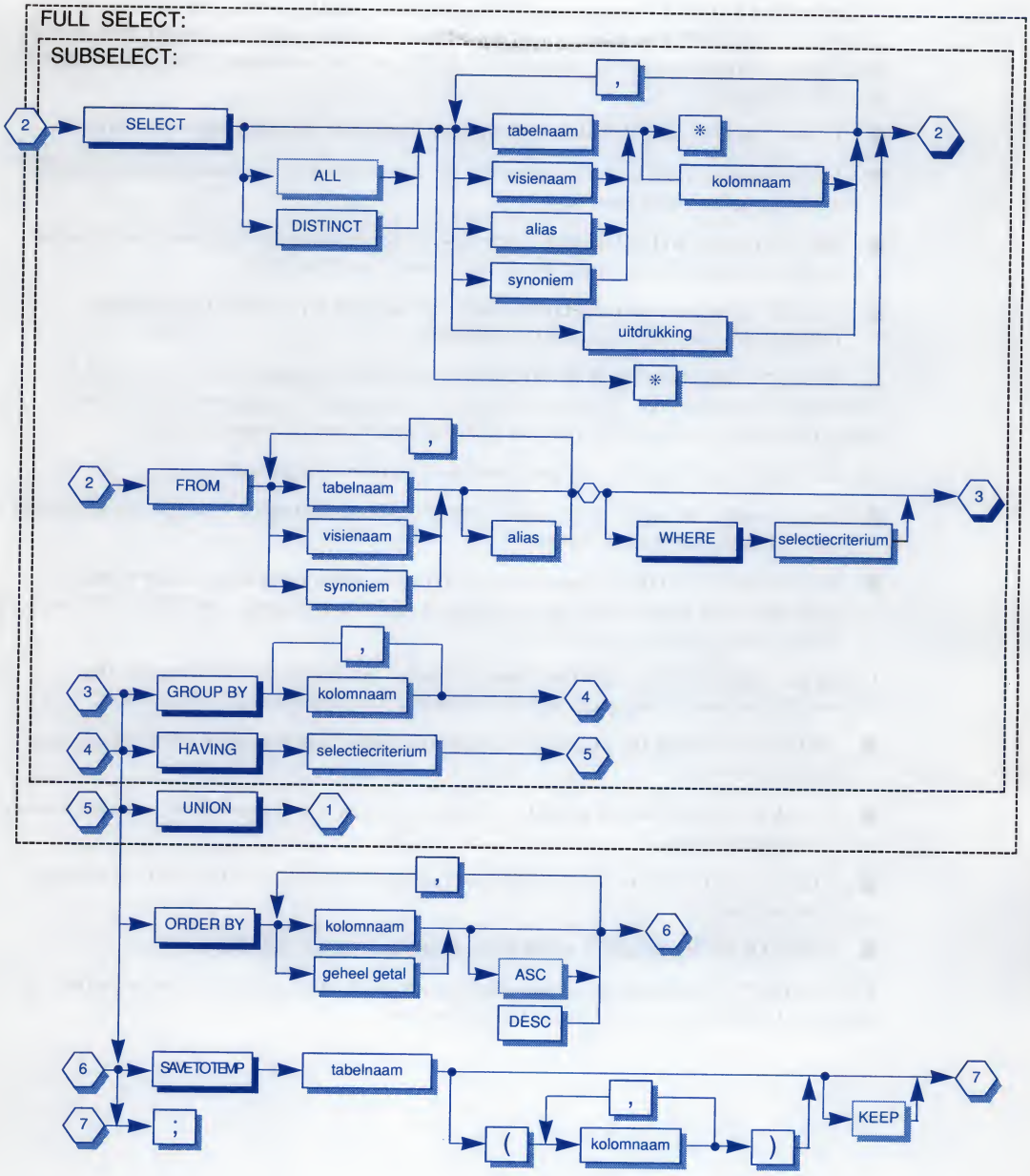
Met de volgende syntaxis kunt u het SELECT-commando in de ingebedde en de interactieve modus gebruiken om gegevens uit tabellen en visies op te halen en weer te geven. De uitgebreide SELECT-syntaxis voor bewerkingen waarmee gegevens worden opgehaald is als volgt:

```
SELECT [ALL/DISTINCT] <kolommenlijst>
    FROM <tabelnaam>/<visienaam> [alias]
        [,<tabelnaam>/<visienaam> [alias]...]
    [WHERE [NOT]<selectiecriterium>
        [AND/OR[NOT] <selectiecriterium>...]]
    [GROUP BY <kolomnaam> [,<kolomnaam>...]]
    [HAVING [NOT]<selectiecriterium>
        [AND/OR[NOT] <selectiecriterium>...]]
    [ORDER BY <kolomnaam>/<geheel getal> [ASC/DESC]
        [,<kolomnaam>/<geheel getal> [ASC/DESC]...]]
    [SAVE TO TEMP <tabelnaam>
        [(<kolomnaam> [,<kolomnaam>...]) [KEEP]];
```

Wanneer bij het ophalen de clausule UNION wordt gebruikt, is de syntaxis als volgt:

```
<SELECT-commando> [UNION <SELECT-commando>...]
    [ORDER BY <kolomnaam>/<geheel getal> [ASC/DESC]
            [,<kolomnaam>/<geheel getal> [ASC/DESC]...]]
    [SAVE TO TEMP <tabelnaam>
            (<kolomnaam> [,<kolomnaam>...]) [KEEP]];
```

OPMERKING
Voordat u een clausule FOR UPDATE OF in een ingebed SQL-programma gebruikt, kunt u deze eerst in een interactief SELECT-commando gebruiken om de resultaten van een query te testen. Als een SELECT-commando de clausule ORDER BY bevat, is de clausule FOR UPDATE OF niet toegestaan.



Afbeelding 6-30 SELECT voor alleen weergave

SELECT

U kunt met het SELECT-commando gegevens van meer tabellen *samenvoegen*. Met een dergelijke zogenaamde join kunt u gegevens uit meer tabellen weergeven zonder daarvoor een programma te hoeven schrijven. Voor het SELECT-commando waarmee u een join definieert, geldt het volgende:

- U moet van elke tabel de kolommen opgeven die u in de resultaattabel wilt weergeven.
- U kunt met een asterisk (*) aangeven dat alle kolommen van alle samengevoegde tabellen moeten worden weergegeven.
- Met de clausule WHERE kunt u *voorwaarden voor samenvoegen* opgeven, waarmee het aantal rijen dat in het resultaat wordt opgenomen, wordt beperkt.
- Voor het groeperen kunt u gebruik maken van GROUP BY en HAVING, terwijl u ORDER BY voor het sorteren kunt gebruiken.

Een SELECT-commando kan in de WHERE- of HAVING-clausule van een ander SELECT-commando worden *genest*. Zo'n genest SELECT-commando, de zogenaamde *sub-query* of *lagere* query, levert de waarden voor het selectiecriterium van de *hogere* query op.

SELECT wordt voor de sub-query *enkelvoudig* of *gecorreleerd* gebruikt:

- Enkelvoudig - De lagere query wordt eerst uitgevoerd waarna het resultaat wordt gebruikt om de hogere query uit te voeren.
- Gecorreleerd - De hogere query wordt eerst uitgevoerd waarna het resultaat wordt gebruikt om de lagere query uit te voeren. De lagere query wordt voor iedere rij die de hogere query oplevert, uitgevoerd.

U kunt het SELECT-commando niet alleen gebruiken om gegevens weer te geven. Het commando kan ook als sub-query worden gebruikt in de volgende SQL-commando's:

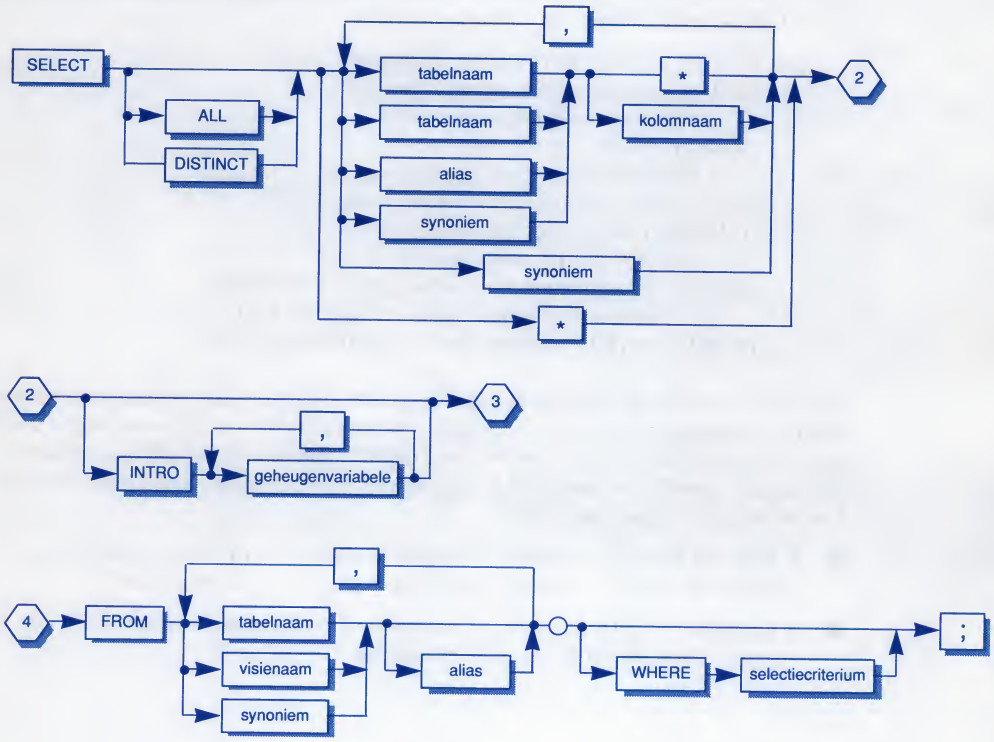
- DELETE en UPDATE: SELECT vormt een onderdeel van het met WHERE opgegeven selectiecriterium.
- INSERT: SELECT wordt gebruikt voor het selecteren van de rijen die in een tabel moeten worden ingevoegd.
- SELECT: SELECT vormt een onderdeel van het met WHERE of HAVING opgegeven selectiecriterium.
- CREATE VIEW: SELECT wordt gebruikt om een visie te definiëren.

In het SELECT-commando dat als sub-query wordt gebruikt, mogen de clausules FROM, WHERE, GROUP BY en HAVING zijn opgenomen.

Ingebed SELECT voor een enkele rij (met INTO)

Het commando SELECT levert in deze vorm één resultaatrij op. U kunt in ingebedde SQL-modus gebruik maken van het SELECT-commando in combinatie met de clausule INTO. Daarmee kunnen waarden uit de rij van een resultaattabel naar dBASE-geheugenvariabelen worden overgebracht. De uitgebreide syntaxis voor deze vorm van het SELECT-commando is als volgt:

```
SELECT [ALL/DISTINCT] <kolommenlijst>
    INTO <lijst geheugenvariabelen>
    FROM <tabelnaam>/<visienaam> [alias]
        [,<tabelnaam>/<visienaam> [alias]...]
    [WHERE [NOT]<selectiecriterium>
        [AND/OR [NOT] <selectiecriterium...]];
```



Afbeelding 6-31 Ingebed SELECT voor een enkele rij (met INTO)

OPMERKING

Wanneer een SELECT-commando de clausule FOR UPDATE OF, GROUP BY, HAVING, UNION, ORDER BY of SAVE TO TEMP bevat, kunt u niet de clausule INTO gebruiken. Levert het SELECT-commando meer dan een rij op, dan worden alleen de waarden uit de eerste rij naar de bij INTO vermelde geheugenvariabelen overgebracht.

De waarden uit de enkele resultaatrij worden afzonderlijk en van links naar rechts overgedragen aan de variabelen die bij de clausule INTO zijn opgegeven. Geheugenvariabelen die niet bestaan, worden tijdens de overdracht gemaakt en krijgen het gegevenstype van de corresponderende rijwaarden.

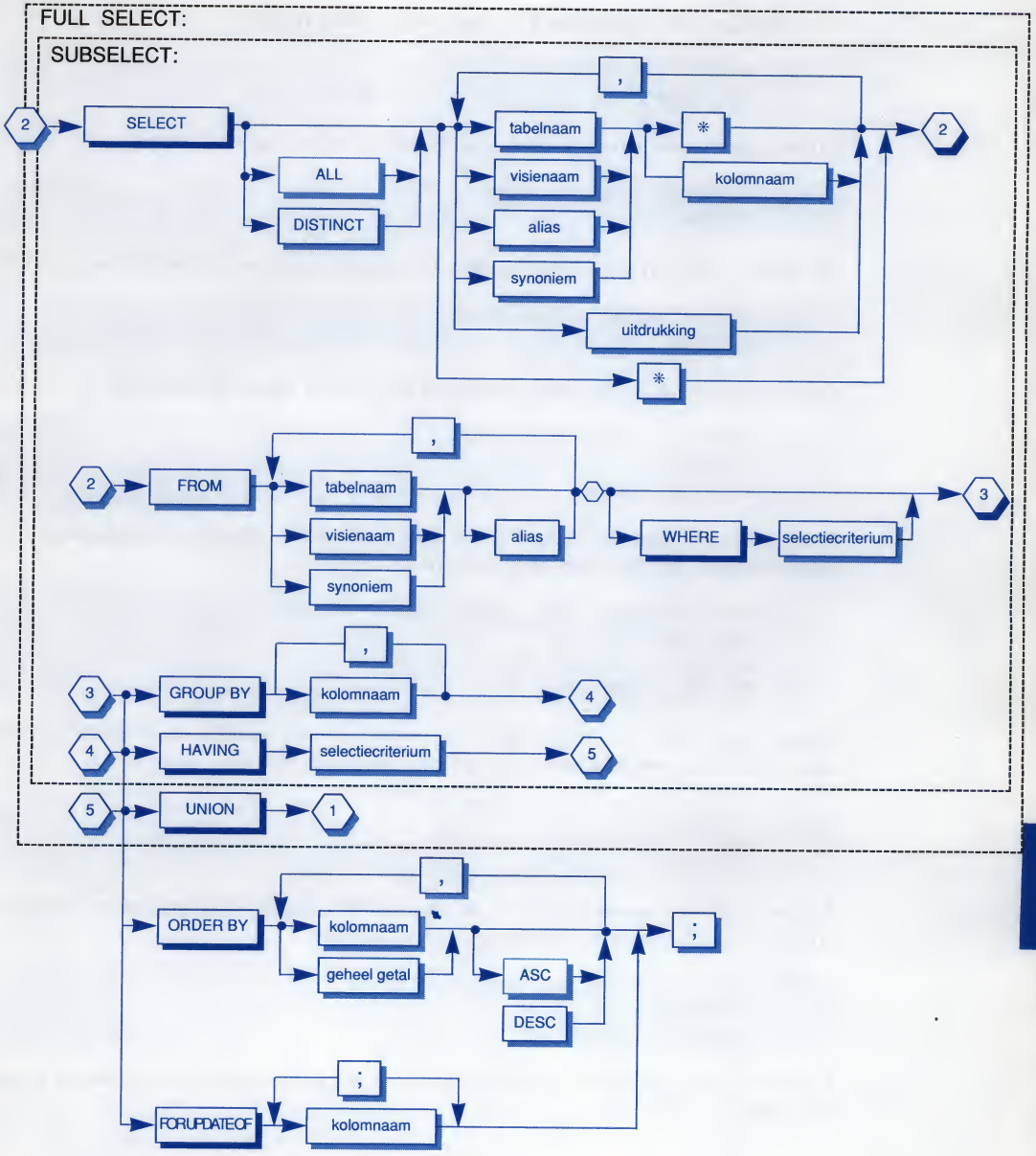
Ingebed SELECT (gebruikt met DECLARE CURSOR)

In een ingebed SQL-programma kunt u met het SELECT-commando dat bij het commando DECLARE CURSOR is gedefinieerd, een resultaattabel maken. De uitgebreide syntaxis voor het SELECT-commando van een cursor is als volgt:

```
SELECT [ALL/DISTINCT] <kolommenlijst>/*
    FROM <tabelnaam>/<visienaam> [alias]
        [,<tabelnaam>/<visienaam> [alias]...]
    [WHERE [NOT]<selectiecriterium>
        [AND/OR[NOT] <selectiecriterium>...]]
    [GROUP BY <kolomnaam> [,<kolomnaam>...]]
    [HAVING [NOT]<selectiecriterium>
        [AND/OR[NOT] <selectiecriterium>...]]
    [ORDER BY <kolomnaam>/<geheel getal> [ASC/DESC]
        [,<kolomnaam>/<geheel getal> [ASC/DESC]...]]
    [FOR UPDATE OF <kolomnaam> [,<kolomnaam>...]];
```

Nadat u het commando OPEN hebt gegeven, wordt de cursor geactiveerd en het bijbehorende SELECT-commando uitgevoerd. Vervolgens wordt de cursor met het commando FETCH achtereenvolgens op alle rijen van de door SELECT gemaakte resultaattabel geplaatst. Vervolgens worden de waarden uit de rijen aan dBASE-geheugenvariabelen overgedragen. U kunt er dan het volgende mee doen:

- U kunt met UPDATE de kolomwaarden bijwerken in de rij van de basistabel die correspondeert met de door de cursor aangewezen rij van de resultaattabel.
- U kunt met DELETE de rij van de basistabel verwijderen die correspondeert met de door de cursor aangewezen rij van de resultaattabel.



Afbeelding 6-32 Ingebed SELECT (gebruikt met DECLARE CURSOR)

Voorbeelden

Om alle kolommen van de tabel Verkopen weer te geven typt u:

```
SQL. SELECT *
     FROM Verkopen;
```

Om bepaalde kolommen uit de tabel Klanten weer te geven typt u:

```
SQL. SELECT Bedrijf, Voornaam, Achtnaam
     FROM Klanten;
```

Om alleen unieke rijwaarden van bepaalde kolommen uit de tabel Invent weer te geven typt u:

```
SQL. SELECT DISTINCT Art_nr, Omschr, Eenhpr
     FROM Invent;
```

Om de gegevens uit geselecteerde rijen uit de tabel Invent weer te geven typt u:

```
SQL. SELECT Art_nr, Omschr, Plaats, Voorraad
     FROM Invent
     WHERE Plaats = "Zwolle";
```

Om gegevens uit rijen van de tabel Invent weer te geven die voldoen aan verscheidene selectiecriteria (tussen haakjes gegroepeerd) typt u:

```
SQL. SELECT Art_nr, Omschr, Plaats, Voorraad, Eenhpr
     FROM Invent
     WHERE (Plaats = "Groningen" OR Plaats = "Middelburg")
     AND (Voorr < 20 AND Eenhpr < 1000);
```

Om gewone waarden uit de tabel Medew te combineren met waarden uit rekenkolommen en een waarde die is gemaakt door een constante uitdrukking typt u:

```
SQL. SELECT Medew_nr, Voornaam, Achtnaam,
     Salaris*12, "(Jaarsalaris)"
     FROM Medew;
```

U kunt de dBASE-functie DATE() in een clausule WHERE als selectiecriterium gebruiken om rijen uit de tabel Medew te selecteren. U moet dan typen:

```
SQL. SELECT Voornaam, Achtnaam, Indienst
     FROM Medew
     WHERE (DATE()-Indienst) > 365*5;
```

U kunt de rijen van Invent in de resultaattabel met de clausule ORDER BY sorteren. U moet dan typen:

```
SQL. SELECT Art_nr, Omschr, Plaats, Voorraad
     FROM Invent
     ORDER BY Omschr ASC;
```

U kunt voor het maken van de resultaattabel van de tabel Klanten een verzamelfunctie gebruiken. U moet dan typen:

```
SQL. SELECT COUNT(*)
     FROM Klanten
     WHERE Prov = "Frsld";
```

U kunt de rijen in de resultaattabel van de tabel Invent met GROUP BY op basis van het Art_nr groeperen. U moet dan typen:

```
SQL. SELECT Art_nr, SUM(Voorraad)
     FROM Invent
     GROUP BY Art_nr;
```

Om gegevens uit de tabel Verkopen en uit de tabel Medew met een join samen te voegen typt u:

```
SQL. SELECT Opdr_nr, Verk_dat, Medew.Medw_nr,  Achtnaam
     FROM Verkopen, Medew
     WHERE Verkopen.Medew_nr = Medew.Medew_nr;
```

U kunt gegevens in een nieuwe tabel invoegen en daarbij het SELECT-commando gebruiken. U moet dan typen:

```
SQL. INSERT INTO Amst
     SELECT *
     FROM Medew
     WHERE Standpl = "Amsterdam";
```

Zie ook

DECLARE CURSOR, DELETE, FETCH, INSERT, UPDATE

SHOW DATABASE

Met het commando SHOW DATABASE kunt u gegevens over iedere gewenste database weergeven. Voor het gebruik van het commando hoeft een database niet actief te zijn.

Syntaxis

SHOW DATABASE;

Afbeelding 6-33 Het commando SHOW DATABASE

Gebruik

Met het commando SHOW DATABASE kunt u de volgende gegevens over iedere gewenste database weergeven:

- de database-naam;
- de gebruikersnaam van degene die de database heeft gemaakt;
- de datum waarop de database is gemaakt;
- het DOS-pad van de database-bestandsindex.

Al deze gegevens zijn opgeslagen in de hoofdcatalogustabel, Sysdbs, die zich in de dBASE IV SQL-bestandsindex (SQLHOME) bevindt.

OPMERKING
De regel SQLHOME = *in het bestand Config.db geeft het pad voor de SQL-bestandsindex op. Standaard is de regel op* SQLHOME = C:\DBASE\SQLHOME *ingesteld.*

Voorbeeld

Om gegevens over alle beschikbare databases weer te geven typt u:

```
SQL. SHOW DATABASE;
```

Zie ook

CREATE DATABASE, DROP DATABASE, START DATABASE, STOP DATABASE

START DATABASE

Met het commando START DATABASE wordt een database geactiveerd. Alle daarna gegeven commando's hebben op die actieve database betrekking.

Syntaxis

START DATABASE [<database-naam>];

Afbeelding 6-34 Het commando START DATABASE

Gebruik

Doorgaans is START DATABASE het eerste commando dat u geeft voordat u interactief andere SQL-commando's geeft. Vaak ook is START DATABASE het eerste commando dat in een SQL-programma-bestand wordt uitgevoerd.

Er kan maar één database actief zijn. Om een database te sluiten en een andere actief te maken geeft u het commando START DATABASE met de naam van de nieuwe database. De oude database wordt dan automatisch gesloten en alle bijbehorende cursors en tijdelijke tabellen worden verwijderd. Met het commando STOP DATABASE kunt u een database sluiten zonder een nieuwe database actief te maken.

Wanneer SET CATALOG op ON is ingesteld, worden met START DATABASE alle tabellen in de actief gemaakte database aan het geopende catalogusbestand toegevoegd. De SQL-systeemcatalogi worden niet aan het geopende catalogusbestand toegevoegd.

Door de regel SQLDATABASE = <database-naam> in het bestand Config.db op te nemen geeft u een standaard database op. U kunt deze dan actief maken zonder het commando START DATABASE te geven. De standaard database wordt automatisch actief zodra u met het commando SET SQL ON de SQL-modus opstart.

Wanneer u het commando START DATABASE zonder database-naam geeft of de door u opgegeven database niet kan worden geopend, wordt er een foutmelding weergegeven. In dat geval wordt de standaard database geopend. Als u met CREATE DATABASE een nieuwe database maakt, wordt deze automatisch direct actief.

U kunt het commando START DATABASE zonder database-naam in een SQL-toepassing opnemen als aan de volgende voorwaarden is voldaan:

- Het commando START DATABASE is het eerste commando dat in de toepassing wordt uitgevoerd.
- De inhoud van de bestandsindex waarin de toepassing is ontwikkeld, is in zijn geheel (inclusief catalogustabellen) gekopieerd naar de bestandsindex van waaruit u dBASE IV start (de actieve bestandsindex).

START DATABASE

OPMERKING
De hiervoor vermelde toepassingen worden "enkele-database"-toepassingen genoemd. Deze zijn overdraagbaar, aangezien ze zonder te worden gecompileerd kunnen worden gebruikt op andere computers dan die waarop ze zijn ontwikkeld.

Zie ook

CREATE DATABASE, DROP DATABASE, SHOW DATABASE, STOP DATABASE

STOP DATABASE

Met het commando STOP DATABASE kunt u de actieve database sluiten.

Syntaxis

STOP DATABASE;

Afbeelding 6-35 Het commando STOP DATABASE

Gebruik

Voordat een database met DROP DATABASE kan worden verwijderd, moet deze worden gesloten. U kunt op de volgende manieren een database sluiten:

1. U geeft het commando STOP DATABASE.
2. U geeft het commando START DATABASE om een andere database actief te maken.

Wanneer een database wordt gesloten, worden alle bijbehorende cursors en tijdelijke tabellen verwijderd.

Zie ook

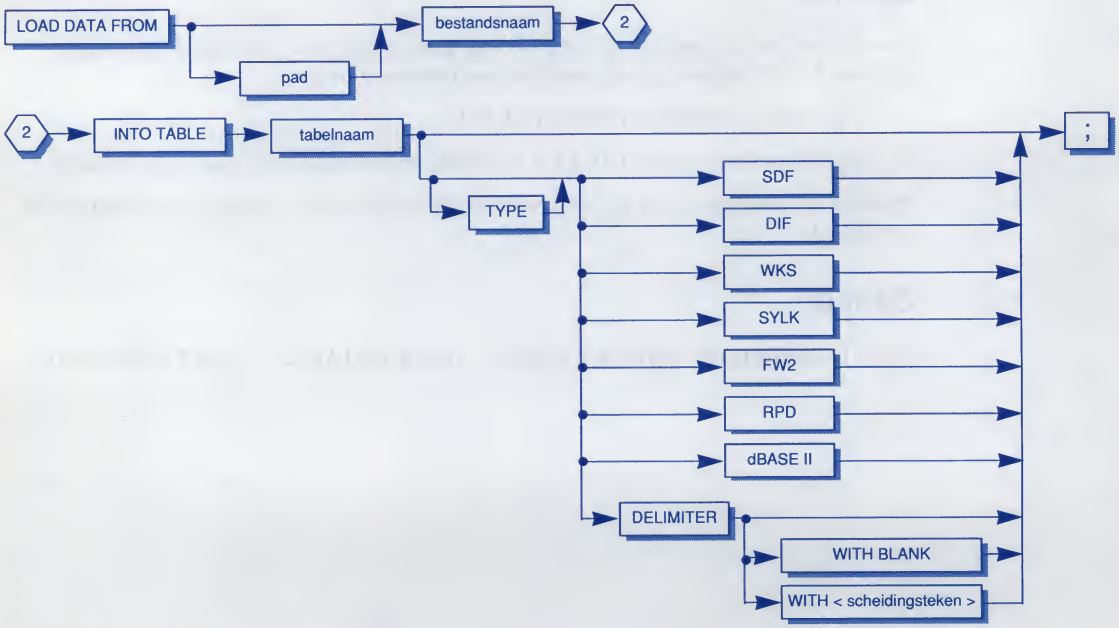
CREATE DATABASE, DROP DATABASE, SHOW DATABASE, START DATABASE

UNLOAD DATA

Met het commando UNLOAD DATA kunt u gegevens uit een SQL-tabel naar een extern bestand exporteren. Wanneer dBASE IV met een wachtwoord is beveiligd, moet u degene zijn die de tabel waaruit gegevens worden geëxporteerd, heeft gemaakt of over het SELECT-recht voor die tabel beschikken.

Syntaxis

UNLOAD DATA TO [pad]<bestandsnaam!
FROM TABLE <tabelnaam!
[[TYPE]SDF/DIF/WKS/SYLK/FW2/RPD/DBASEII/
DELIMITED [WITH BLANK/WITH <scheidingsteken>]];



Afbeelding 6-36 Het commando UNLOAD DATA

Gebruik

Met het commando UNLOAD DATA wordt een nieuw bestand gemaakt waarna gegevens uit een SQL-tabel naar dit bestand worden gekopieerd. Het commando UNLOAD DATA ondersteunt dezelfde bestandsstructuren als het dBASE IV-commando APPEND FROM. U kunt met UNLOAD DATA gegevens exporteren naar de volgende soorten bestanden:

- dBASE II database-bestanden (dBASE II);
- dBASE III, dBASE III PLUS, en dBASE IV database-bestanden;
- RapidFile database-bestanden (RPD);
- Framework II database- en spreadsheet-bestanden (FW2);
- ASCII-bestanden, structuur met scheidingstekens (DELIMITED);
- ASCII-bestanden, structuur "System data" (SDF);
- VisiCalc structuurbestanden (DIF);
- Multiplan spreadsheet-structuurbestanden (SYLK);
- Lotus 1-2-3 structuurbestanden (WKS).

Het commando UNLOAD DATA maakt het bestand waarnaar de gegevens worden geëxporteerd in de actieve bestandsindex, tenzij u een expliciet pad opgeeft. Als u bij TYPE geen bestandstype opgeeft, worden de gegevens naar een database-bestand (.dbf) geëxporteerd. De gegevens worden met UNLOAD DATA altijd naar een ongecodeerd bestand geëxporteerd, ongeacht of dBASE IV met een wachtwoord is beveiligd.

De velden in het bestand worden gemaakt op basis van de namen en definities van de gekopieerde kolommen. Spreadsheet-bestanden gebruiken de kolomnamen van de tabel als koppen voor kolommen.

Met de optie DELIMITED kan worden opgegeven dat de gegevens worden geëxporteerd naar een ASCII-tekstbestand (.txt) met scheidingstekens. Aan het bestand waarnaar wordt geëxporteerd, worden bij UNLOAD DATA (afhankelijk van hetgeen bij DELIMITED wordt opgegeven) de volgende eisen gesteld:

- DELIMITED - De bestandsvelden worden gescheiden door komma's en tekenvelden worden gescheiden door dubbele aanhalingstekens. Deze instelling kan ook worden opgegeven met DELIMITED WITH ".
- DELIMITED WITH BLANK - De bestandsvelden worden gescheiden door een spatie terwijl tekenvelden niet worden gescheiden.
- DELIMITED WITH <scheidingsteken> - De bestandsvelden worden gescheiden door komma's terwijl tekenvelden worden gescheiden door het opgegeven scheidingsteken. Het standaard scheidingsteken is het dubbele aanhalingsteken.

UNLOAD DATA

Voorbeeld

U kunt gegevens uit de tabel Medew exporteren naar een dBASE IV-database-bestand in de huidige bestandsindex. U moet dan typen:

```
SQL. UNLOAD DATA TO C:\BACKUP\Medew2 FROM  TABLE Medew;
```

Zie ook

LOAD DATA

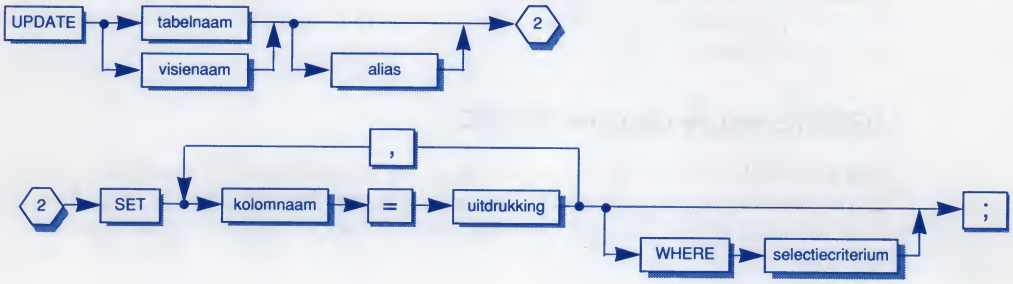
UPDATE

Met het commando UPDATE kunt u kolomwaarden in opgeven rijen van een bewerkbare visie of van een tabel bijwerken (raadpleeg het gedeelte over CREATE VIEW).

Het commando UPDATE kent twee vormen. De eerste variant is zowel in de interactieve als in de ingebedde SQL-modus te gebruiken. De tweede variant wordt alleen in ingebedde SQL-modus met een gedefinieerde cursor gebruikt om de rijen in de basistabel die corresponderen met de rijen die in de resultaattabel door de cursor worden aangewezen, bij te werken.

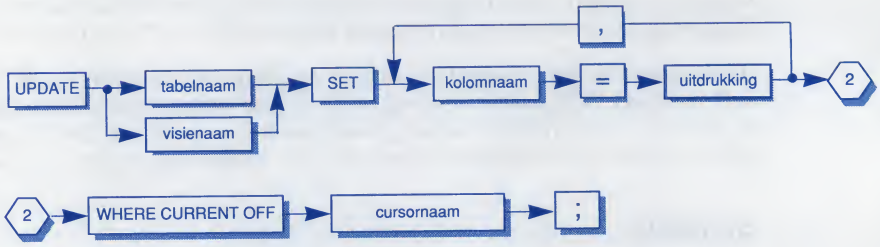
Syntaxis

UPDATE <tabelnaam>/<visienaam>
SET <kolomnaam> = <uitdrukking>
[,<kolomnaam> = <uitdrukking>...]
[WHERE <selectiecriterium>]; (1)



Afbeelding 6-37 Het commando UPDATE

UPDATE <tabelnaam>/<visienaam>
SET <kolomnaam> = <uitdrukking>
[,<kolomnaam> = <uitdrukking>...]
WHERE CURRENT OF <cursornaam>; (2)



Afbeelding 6-38 Het commando UPDATE met de clausule WHERE CURRENT OF

Gebruik

Wanneer dBASE IV met een wachtwoord is beveiligd, moet u degene zijn die de tabel of visie die wordt bijgewerkt, heeft gemaakt, of over het UPDATE-recht beschikken voor ieder kolom die u wilt bijwerken.

UPDATE met de clausule WHERE

Met de clausule WHERE kunt u rijen selecteren die u wilt bijwerken. De clausule WHERE kan bestaan uit een selectiecriterium of uit een sub-query. Een sub-query die in de clausule WHERE wordt gebruikt, mag niet verwijzen naar de tabel of visie die u op dat moment bijwerkt.

OPMERKING
Als u een visie bijwerkt die met de optie WITH CHECK OPTION is gedefinieerd, moeten de bijgewerkte kolomwaarden voldoen aan de voorwaarden die zijn opgegeven bij de clausule WHERE waarmee de visie is gedefinieerd.

UPDATE met de clausule WHERE CURRENT OF

De clausule WHERE CURRENT OF wordt in een ingebed SQL-programma gebruikt als u de rijen uit de basistabel wilt bijwerken die corresponderen met de rijen die in de resultaattabel door de cursor worden aangewezen. De cursor waarnaar in WHERE CURRENT OF wordt verwezen, moet eerder in het programma met DECLARE CURSOR zijn gedefinieerd en met OPEN zijn geopend.

De rij in de basistabel die wordt bijgewerkt, correspondeert met de rij die in de resultaattabel door de cursor wordt aangewezen. In deze rij staan de waarden die het laatst met het commando FETCH zijn bewerkt.

TIP
U kunt controleren of met de clausule WHERE die u bij het commando UPDATE hebt opgegeven, de juiste rijen worden bijgewerkt. U moet de clausule dan eerst in een SELECT-commando gebruiken.

Voorbeelden

In het volgende voorbeeld ziet u hoe de kolom Commissie uit de tabel Medew wordt bijgewerkt met een commissieverhoging voor verkopers die in vaste dienst zijn.

```
SQL.  UPDATE Medew
      SET Commissie = Commissie * 1.25
      WHERE Indienst < {05/07/82};
```

Het volgende programmasegment kan worden gebruikt om de tabel Verkopen bij te werken voor gefactureerde orders.

```
DECLARE Fact CURSOR FOR
    SELECT Order_nr, Verk_dat, Medew_nr, Klant_nr, Gefact
    FROM Verkopen
    WHERE NOT Gefact
    FOR UPDATE OF Gefact;
        .
        .
        .
OPEN Fact;
    DO WHILE .T.
        FETCH Fact INTO morder_nr, mverk_dat, mmedew_nr, mklant_nr,  mgefact;
        IF SQLCODE = 0
        * Druk een factuur voor ordernummer in de huidige rij af
        * Indien gelukt, wijzig geheugenvariabele mgefact in .T.
        DO Factuur WITH mopdr_nr, mverk_dat, mmedew_nr, mklant_nr, mgefact
            IF mgefact
                UPDATE Verkopen
                SET Gefact = .T.
                WHERE CURRENT OF Fact;
            ENDIF
        ELSE
            EXIT
        ENDIF
    ENDDO
CLOSE Fact;
```

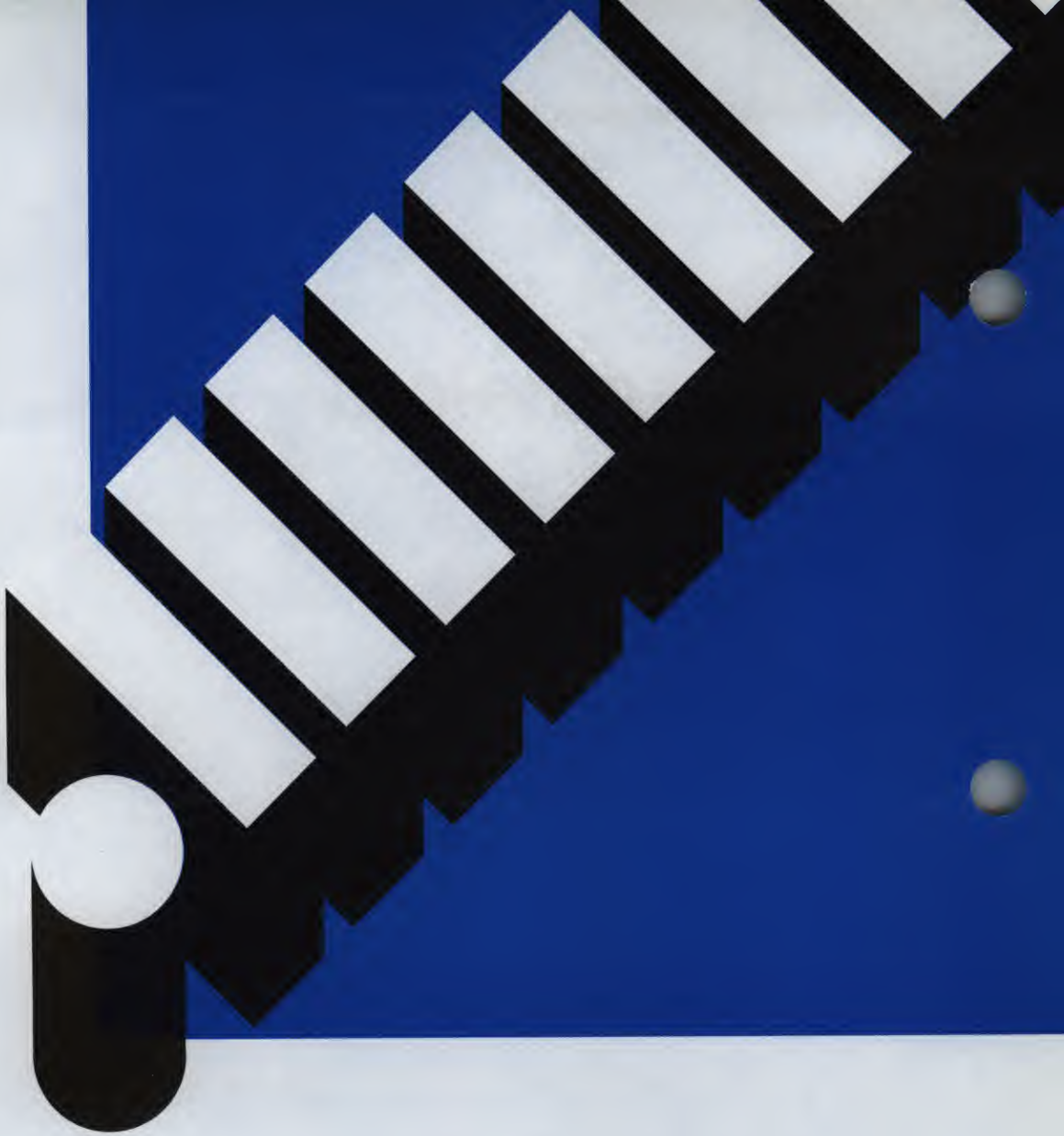
In dit voorbeeld genereert het programma Factuur een factuur voor de order uit de rij die door de cursor wordt aangewezen. Nadat de order is gemaakt, wordt de waarde van Gefact in de actieve rij in waar (.T.) gewijzigd.

Zie ook

DECLARE CURSOR, FETCH, OPEN, SELECT

Naslagshandboek van dBASE IV

SQL-catalogi



i 1 2 3 4 5 6 7 A B

C D E F G H In

SQL-catalogi

dBASE IV SQL legt gegevens over alle *objecten* (tabellen, visies, synoniemen, indexen) in een database met behulp van *catalogustabellen* vast. De gegevens uit deze catalogustabellen worden door SQL voor het uitvoeren van queries en voor het bijwerken gebruikt.

Bij iedere database die u met het commando CREATE DATABASE maakt, wordt een set catalogustabellen geconstrueerd. In de SQL *home*-bestandsindex (volgens de standaardinstelling \DBASE\SQLHOME) wordt een hoofdcatalogustabel (Sysdbds) bijgehouden waarin gegevens over iedere SQL-database worden vastgelegd. De Sysdbs-tabel bevat de naam van alle databases, de gebruikersnaam van degene die de database heeft gemaakt, de datum waarop de database is gemaakt en het volledige pad naar de database. Met het commando SHOW DATABASE kunt u een overzicht opvragen van de databases die in de Sysdbs-tabel zijn opgenomen.

De catalogus van ieder database bestaat uit de volgende 10 SQL-tabellen:

Tabelnaam	Omschrijving
Sysauth	In de tabel Sysauth zijn de rechten van gebruikers voor tabellen en visies vastgelegd. Voor iedere gebruiker en voor iedere tabel of visie waaraan respectievelijk één of meer rechten zijn toegewezen, wordt een rij ingevoegd.
Syscolau	In de tabel Syscolau worden de mutatierechten die gebruikers voor kolommen in een tabel of een bewerkbare visie hebben, vastgelegd. Voor elk gegeven GRANT-commando, voor elke begunstigde gebruiker en voor elke kolom waarvoor UPDATE-recht wordt toegewezen, wordt een rij ingevoegd.
Syscols	In de tabel Syscols worden alle kolommen in alle tabellen en visies (inclusief de catalogustabellen) vastgelegd. Voor iedere kolom (in iedere tabel of visie in de actieve database) wordt een rij ingevoegd.
Sysidxs	In de tabel Sysidxs worden alle indexen en de tabellen waarvoor deze zijn gedefinieerd, vastgelegd. Voor iedere index wordt een rij ingevoegd.
Syskeys	In de tabel Syskeys worden alle kolommen (indexsleutels) uit alle indexen vastgelegd. Voor iedere kolom uit iedere index wordt een rij ingevoegd.
Syssyns	In de tabel Syssyns worden alle synoniemdefinities voor alle tabellen en visies vastgelegd. Voor ieder synoniem dat in de actieve database is gedefinieerd, wordt een rij ingevoegd.
Systabls	In de tabel Systabls worden gegevens over alle tabellen en visies vastgelegd. Voor iedere tabel en visie die in de actieve database is gedefinieerd, wordt een rij ingevoegd.

(wordt vervolgd)

Tabelnaam	Omschrijving
Systimes	In de tabel Systimes worden gegevens met betrekking tot multi-user-omgevingen vastgelegd om ervoor te zorgen dat gebruikers over de meest recente kopie van SQL-catalogustabellen beschikken.
Sysvdeps	In de tabel Sysvdeps wordt de relatie tussen visies en tabellen in de actieve database vastgelegd (zodat visies worden verwijderd als een basistabel van die visie wordt verwijderd). Voor ieder tabel die in de definitie van een visie wordt opgenomen, wordt een rij ingevoegd.
Sysviews	In de tabel Sysviews worden de definities en de gebruiksbeperkingen van iedere visie vastgelegd. Voor iedere definitie van een visie worden één of meer rijen ingevoegd.

OPMERKING
De catalogustabellen die in SQL worden gebruikt, lijken op de tabellen die voor IBM DB2-database zijn gedefinieerd. Onderlinge verschillen zijn te wijten aan verschillen tussen de mainframe- en microcomputer-omgevingen.

U kunt gegevens uit een systeemcatalogustabel op dezelfde manier als gegevens uit andere SQL-tabellen met het commando SELECT weergeven. Om gegevens over alle tabellen en visies in de actieve database uit de tabel Systables weer te geven typt u:

```
SQL. SELECT Tbname, Creator, Tbtype
FROM Systables
```

Door dit commando te geven worden gegevens over tabellen en visies in de actieve database weergegeven. In de kolom Tbtype staat aangegeven of het bestand een tabel (T) of een visie (V) is. Met de letters D en K wordt aangegeven of een met het commando SELECT...SAVE TO TEMP gemaakte tabel tijdelijk of permanent is (K betekent dat de optie KEEP is gebruikt).

U kunt de inhoud van catalogustabellen weliswaar bekijken, maar u kunt deze niet bijwerken.

OPMERKING
De gebruiker met de SQLDBA-gebruikersnaam heeft bijzondere rechten en kan catalogustabellen met het commando UPDATE direct bijwerken. Hij kan echter niet de structuur van de SQL-catalogustabellen met het commando ALTER wijzigen.

Catalogustabellen bijwerken

De catalogustabellen bevatten twee soorten gegevens, te weten SQL-objectdefinities en statistieken met betrekking tot opgeslagen gegevens. Met de SQL-objectdefinities worden SQL-objecten (databases, tabellen, visies, synoniemen en indexen) gespecificeerd. De objectdefinities worden automatisch bijgewerkt wanneer er SQL-commando's worden uitgevoerd. De SQL-catalogustabellen worden automatisch bijgewerkt wanneer u een database maakt of er objecten aan toevoegt en wanneer u toegangsrechten voor tabellen en visies toevoegt of wijzigt. U kunt de in catalogustabellen vastgelegde gegevens van specifiek opgegeven tabellen bijwerken met het commando CREATE INDEX of DBDEFINE.

Wanneer u de gegevens in tabellen wijzigt, bijvoorbeeld met het commando INSERT, UPDATE of DELETE, worden de catalogustabellen niet automatisch bijgewerkt. Aangezien SQL voor een optimaal prestatievermogen de statistische informatie over tabellen en gerelateerde indexen gebruikt, moet u regelmatig het commando RUNSTATS uitvoeren.

U moet het commando RUNSTATS in elk geval uitvoeren wanneer u de definities van een tabel hebt gewijzigd, wanneer u met het commando DBDEFINE een .dbf-bestand en een bijbehorend .mdx-bestand hebt bewerkt, of wanneer u gegevens in meer dan tien procent van de rijen in een tabel hebt gewijzigd. Wanneer u het commando RUNSTATS uitvoert, moet u geen andere database-bewerkingen uitvoeren. Raadpleeg voor meer informatie het gedeelte over RUNSTATS in hoofdstuk 6.

Met het commando RUNSTATS worden de catalogi vergeleken met de actuele database-tabellen en visies en met mogelijke ander objecten in de database. Bij verschillen wordt de catalogus bijgewerkt. Nadat RUNSTATS is uitgevoerd, verschijnt de melding **RUNSTATS geslaagd** wanneer alle verschillen zijn bijgewerkt. Mocht zich een probleem hebben voorgedaan, dan verschijnt de melding **RUNSTATS voltooid zonder wijziging van catalogus; fout/waarschuwing gevonden**. Ga in dat geval te werk volgens de procedure die voorafgaand aan de RUNSTATS-melding wordt gesuggereerd en voer het commando opnieuw uit.

Door RUNSTATS worden de volgende kolommen bijgewerkt:

Kolom	Tabel
COLCARD	Syscols
HIGH2KEY	Syscols
LOW2KEY	Syscols
FIRSTKCARD	Sysidxs
FULLKCARD	Sysidxs
NLEAF	Sysidxs
NLEVEL	Sysidxs
UPDATED	Systabls
CARD	Systabls
NPAGES	Systabls

OPMERKING
De kolommen COLCARD, HIGH2KEY en LOW2KEY worden alleen voor kolommen van het numerieke gegevenstype (Decimal, Float, Integer, Numeric, Smallint) bijgewerkt.

Omschrijving van catalogustabellen

In het volgende gedeelte worden alle SQL-catalogustabellen en de gegevens die in de kolommen van die tabellen worden opgeslagen, omschreven.

De tabel Sysauth

In de tabel Sysauth staan gegevens over de rechten voor tabellen en visies die aan bepaalde gebruikers zijn toegewezen. Deze tabel wordt bijgewerkt door het uitvoeren van de commando's GRANT en REVOKE. Voor iedere toewijzing van rechten voor een bepaalde tabel aan een bepaalde gebruiker (met PROTECT toegewezen gebruikersnaam) wordt een rij van 77 bytes ingevoegd. Het aantal rijen dat per succesvol uitgevoerd GRANT-commando wordt ingevoegd, is het produkt van het aantal begunstigden in de lijst bij TO en het aantal tabellen in de lijst bij ON.

Kolom	Type	Omschrijving
GRANTOR	Char(10)	Gebruikersnaam van degene die rechten toewijst.
TNAME	Char(10)	Naam van de tabel waarvoor de rechten zijn toegewezen
USERID	Char(10)	Gebruikersnaam van degene aan wie rechten worden toegewezen
TBTYPE	Char(1)	De kolom TBTYPE bevat de waarde T voor tabellen en de waarde V voor visies. De waarden D en K specificeren tijdelijke tabellen die met SELECT...SAVE TO TEMP zijn gemaakt (K geeft daarbij aan dat de optie KEEP is gebruikt).
PSELECT	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
PINSERT	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
PDELETE	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
PALTER	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
PINDEX	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
PUPDATE	Char(4)	Bevat het bereik van de toegewezen UPDATE-rechten: ALL (alle kolommen), SOME (opgegeven kolommen), NONE (wanneer geen rechten zijn toegewezen).
GRANTOPT	Char(1)	Wanneer de GRANT-optie is gebruikt (de gebruiker kan in dat geval het recht aan derden toewijzen), bevat de kolom GRANTOPT de waarde G. Indien de optie niet is gebruikt, bevat de kolom de waarde Y.

De tabel Syscolau

In de tabel Syscolau staan gegevens over de UPDATE-rechten van alle gebruikers met betrekking tot alle kolommen in tabellen of visies in de actieve database. De kolom Syscolau wordt bijgewerkt door het uitvoeren van de commando's GRANT en REVOKE. Voor iedere GRANT-bewerking waarmee het mutatierecht voor een kolom aan een begunstigde (opgegeven met specifieke gebruikersnaam of met PUBLIC) wordt toegewezen, wordt per gebruiker een rij van 50 bytes per kolom ingevoegd. Het aantal rijen dat per succesvol uitgevoerd GRANT-commando wordt ingevoegd, is het produkt van het aantal begunstigden dat in de lijst bij TO staat vermeld en het aantal kolommen waarvoor het UPDATE-recht wordt verleend.

Kolom	Type	Omschrijving
GRANTOR	Char(10)	Gebruikersnaam van degene die de UPDATE-rechten toewijst (of de visie heeft gemaakt).
TNAME	Char(10)	Naam van de tabel of visie waarvoor het recht is toegewezen.
GRANTEE	Char(10)	Gebruikersnaam van degene aan wie het UPDATE-recht wordt toegekend.
COLNAME	Char(10)	Naam van de kolom waarvoor UPDATE-recht is toegewezen.
AUTHTIME	Numeric(8)	Tijdaanduiding in gehele getallen van de datum waarop het recht is toegewezen.
GRANTOPT	Char(1)	Wanneer de GRANT-optie is gebruikt (de gebruiker kan in dat geval het recht aan derden toewijzen), staat in de kolom GRANTOPT de waarde G. Indien de optie niet is gebruikt, bevat de kolom de waarde Y.

De tabel Syscols

De tabel Syscols bevat omschrijvingen van alle tabellen uit alle kolommen en visies in de actieve database. Voor iedere kolom die aan een tabel (inclusief de catalogustabellen) of visie in de database wordt toegevoegd, wordt een rij van 68 bytes in de tabel Syscols ingevoegd. Wanneer er nieuwe tabellen of visies worden gemaakt of wanneer bestaande tabellen met ALTER worden bewerkt om extra kolommen toe te voegen, worden automatisch nieuwe rijen in Syscols ingevoegd.

Kolom	Type	Omschrijving
COLNAME	Char(10)	Kolomnaam
TBNAME	Char(10)	Naam van de tabel of visie waarvan de kolom deel uitmaakt.
TBCREATOR	Char(10)	Gebruikersnaam van degene die de tabel heeft gemaakt
COLNO	Numeric(3)	Positie (geteld van links naar rechts, beginnend met 1) van de kolom binnen de tabel of visie.

(wordt vervolgd)

Kolom	Type	Omschrijving
COLTYPE	Char(1)	dBASE-gegevenstype van de kolom: C (teken), N (numeriek), D (datum), F (zwevend) of L (logisch); SQL-gegevenstypen Smallint, Integer, Decimal en Numeric worden opgeslagen als gegevenstype N.
COLLEN	Numeric(3)	Kolombreedte
COLSCALE	Numeric(2)	Aantal decimaalposities
NULLS	Char(1)	Gereserveerd voor toekomstig gebruik
COLCARD	Numeric(10)	Aantal verschillende waarden in de kolom (voor kolommen waarbij COLTYPE de waarde N (numeriek), F (zwevend) of D (datum) heeft, wordt COLCARD bijgewerkt door het uitvoeren van CREATE INDEX en RUNSTATS).
UPDATES	Char(1)	Als een kolom kan worden bijgewerkt, bevat de kolom UPDATES de waarde Y. Indien een kolom niet kan worden bijgewerkt, bevat UPDATES de waarde N.
HIGH2KEY	Char(8)	De op één na hoogste waarde van de kolom wanneer deze deel uitmaakt van een index (voor kolommen waarbij COLTYPE de waarde N (numeriek), F (zwevend) of D (datum) heeft, wordt HIGH2KEY bijgewerkt door het uitvoeren van CREATE INDEX en RUNSTATS).
LOW2KEY	Char(8)	De op één na laagste waarde van de kolom wanneer deze deel uitmaakt van een index (voor kolommen waarbij COLTYPE de waarde N (numeriek), F (zwevend) of D (datum) heeft, wordt LOW2KEY bijgewerkt door het uitvoeren van CREATE INDEX en RUNSTATS).

De tabel Sysdbs

De tabel Sysdbs is de *hoofd*catalogustabel waarin gegevens over alle met CREATE DATABASE gemaakte databases worden opgeslagen. Het verschil tussen de hoofdcatalogustabel en de andere catalogustabellen is gelegen in het feit dat er maar één hoofdcatalogustabel is, terwijl voor iedere database een set van alle andere catalogustabellen wordt gemaakt. De hoofdcatalogustabel Sysdbs bevindt zich in de *home*-bestandsindex van SQL (standaard bestandsindex van SQL: \DBASE\SQLHOME). Voor iedere nieuw gemaakte SQL-database wordt een rij van 91 bytes ingevoegd.

Kolom	Type	Omschrijving
NAME	Char(8)	Naam van de database
CREATOR	Char(10)	Gebruikersnaam van degene die de database heeft gemaakt.
CREATED	Date	Datum waarop de database is gemaakt.
PATH	Char(64)	Het DOS-pad van de database-bestandsindex

De tabel Sysidxs

In de tabel Sysidxs worden alle indexen en de tabellen waarvoor deze zijn gedefinieerd, beschreven. Voor iedere nieuwe index die in de actieve database wordt gedefinieerd, wordt een rij van 65 bytes in Sysidxs ingevoegd.

Kolom	Type	Omschrijving
IXNAME	Char(10)	Indexnaam
CREATOR	Char(10)	Gebruikersnaam van degene die de index heeft gemaakt.
TBNAME	Char(10)	Naam van de tabel waarvoor de index is gedefinieerd.
TBCREATOR	Char(10)	Gebruikersnaam van degene die de tabel heeft gemaakt (mogelijk een andere naam dan bij CREATOR).
UNIQUERULE	Char(1)	Als de optie UNIQUE is opgegeven, bevat de kolom UNIQUERULE de waarde U. Indien de optie UNIQUE niet is opgegeven, bevat de kolom de waarde D (om aan te geven dat dubbele waarden zijn toegestaan).
COLCOUNT	Numeric(3)	Aantal kolommen in de indexsleutel.
CLUSTERING	Char(1)	Gereserveerd voor toekomstig gebruik
CLUSTERED	Char(1)	Statistische ingave
FIRSTKCARD	Numeric(5)	Statistische ingave
FULLKCARD	Numeric(5)	Statistische ingave
NLEAF	Numeric(5)	Statistische ingave
NLEVELS	Numeric(3)	Statistische ingave

De tabel Syskeys

In de tabel Syskeys worden alle kolommen omschreven die als indexsleutel zijn gedefinieerd. Voor iedere kolom die in de indexsleutel van een index in de actieve database wordt opgegeven, wordt een rij van 38 bytes in de tabel Syskeys ingevoegd.

Kolom	Type	Omschrijving
IXNAME	Char(10)	Indexnaam
IXCREATOR	Char(10)	Gebruikersnaam van degene die de index heeft gemaakt.
COLNAME	Char(10)	Naam van de kolom die de index definieert.
COLNO	Numeric(3)	Positie van de kolom (geteld van links naar rechts) binnen de tabel.
COLSEQ	Numeric(3)	Relatieve positie van de indexsleutel binnen een indexbestand (.mdx).
ORDERING	Char(1)	Afhankelijk van de sorteervolgorde bevat de kolom ORDERING de waarde A (ASC: oplopend) of D (DESC: afdalend).

De tabel Syssyns

De tabel Syssyns bevat de omschrijvingen van alle synoniemen die in de actieve database zijn gedefinieerd. Voor ieder nieuw synoniem voor een visie- of tabelnaam wordt een rij van 41 bytes ingevoegd.

Kolom	Type	Omschrijving
SYNAME	Char(10)	Synoniem
CREATOR	Char(10)	Gebruikersnaam van degene die het synoniem heeft gemaakt.
TBNAME	Char(10)	Naam van tabel of visie waarvoor het synoniem is gedefinieerd.
TBCREATOR	Char(10)	Gebruikersnaam van degene die de tabel of visie waarvoor het synoniem is gedefinieerd, heeft gemaakt.

De tabel Systables

De tabel Systables bevat omschrijvingen van tabellen en visies die in de actieve database zijn gedefinieerd. Voor iedere nieuwe tabel of visie die wordt gedefinieerd, wordt een rij van 74 bytes ingevoegd.

Kolom	Type	Omschrijving
TBNAME	Char(10)	Naam van de tabel of visie
CREATOR	Char(10)	Gebruikersnaam van degene die de tabel of visie heeft gemaakt.
TBTYPE	Char(1)	De kolom TBTYPE bevat de waarde T als het object een basistabel is, de waarde V als het object een visie is, en de waarde D of K als het object een tabel is die met het commando SELECT...SAVE TO TEMP is gemaakt (de K geeft aan dat de optie KEEP is gebruikt).
COLCOUNT	Numeric(3)	Geeft het aantal kolommen in de tabel of visie aan.
CLUSTERRID	Numeric(10)	Gereserveerd voor toekomstig gebruik
INDXCOUNT	Numeric(3)	Geeft het aantal indexen aan dat op een basistabel is gebaseerd. Voor visies is 0 ingesteld.
CREATED	Date	Datum waarop de tabel of visie is gemaakt.
UPDATED	Date	Datum waarop voor het laatst het commando CREATE INDEX of RUNSTATS is uitgevoerd. In het geval van eenvisie is deze datum dezelfde als die bij CREATED.
CARD	Numeric(10)	Statistische ingave
NPAGES	Numeric(10)	Statistische ingave

De tabel Systimes

De tabel Systimes bevat gegevens die alleen in multi-user-omgevingen worden gebruikt. Met deze gegevens wordt ervoor gezorgd dat de de kopieën van tabellen waarover gebruikers beschikken, met de meest recente wijzigingen van de catalogustabellen zijn bijgewerkt. Voor iedere catalogustabel wordt een rij van 27 bytes bijgehouden (er worden nooit nieuwe rijen ingevoegd). Telkens wanneer een catalogustabel in de actieve database wordt bijgewerkt, worden de kolommen DATEUP en TIMEUP bijgewerkt.

Kolom	Type	Omschrijving
NAME	Char(10)	Naam van de catalogustabel
DATEUP	Date	Datum waarop de catalogustabel voor het laatst is bijgewerkt.
TIMEUP	Char(8)	Tijdstip waarop de catalogustabel voor het laatst is bijgewerkt.

De tabel Sysvdeps

De tabel Sysvdeps bevat omschrijvingen van de tabellen en visies waarop een visie is gebaseerd. Voor iedere tabel of visie waarop in de actieve database een visie wordt gebaseerd, wordt een rij van 36 bytes ingevoegd.

Kolom	Type	Omschrijving
VIEWNAME	Char(10)	Visienaam
TBLNO	Numeric(5)	Positie van de tabel of visie (geteld van links naar rechts, beginnend met 1) in de clausule FROM van de definitie van de visie.
TBNAME	Char(10)	Naam van de tabel of visie waarop de visie is gebaseerd.
CREATOR	Char(10)	Gebruikersnaam van degene die de visie heeft gemaakt.

De tabel Sysviews

In de tabel Sysviews staan omschrijvingen van alle visies. Voor ieder visie die in de actieve database wordt gedefinieerd, worden één of meer rijen van 255 bytes ingevoegd. Samengestelde rijen worden opgenomen als de definitie van de visie meer dan 200 bytes in beslag neemt. Bij samengestelde rijen geeft de waarde van de kolom SEQNO aan, welk positienummer de rij ten opzichte van de andere rijen binnen de omschrijving van de visie heeft.

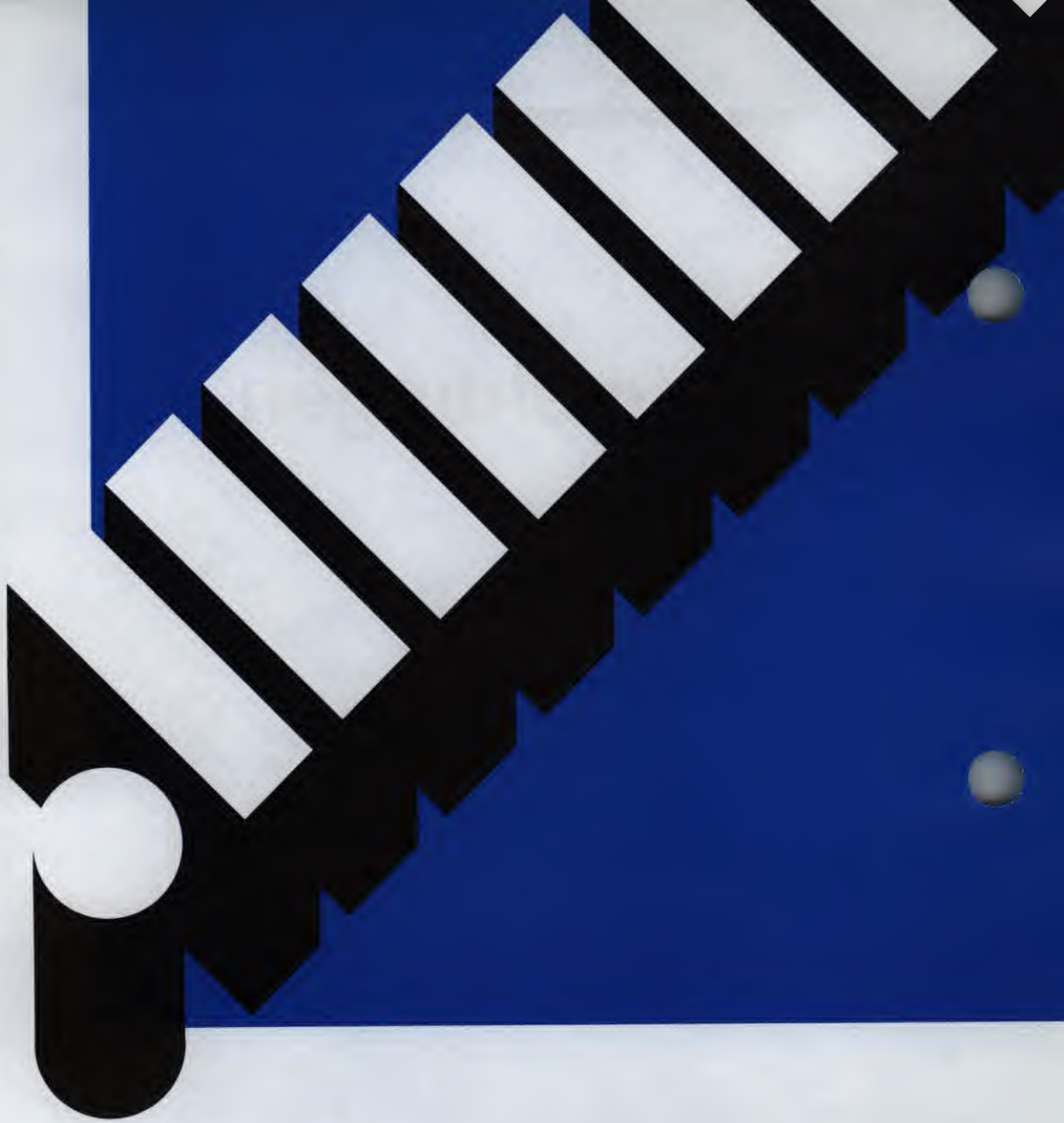
Kolom	Type	Omschrijving
VIEWNAME	Char(10)	Visienaam
CREATOR	Char(10)	Gebruikersnaam van degene die de visie heeft gemaakt.
SEQNO	Numeric(1)	Nummer van een rij (beginnend met 1) ten opzichte van andere rijen uit de omschrijving van dezelfde visie.

(wordt vervolgd)

Kolom	Type	Omschrijving
VCHECK	Char(1)	De kolom VCHECK bevat de waarde Y als de optie CHECK is ingeschakeld. Indien deze optie niet is ingeschakeld, bevat de kolom de waarde N.
READONLY	Char(1)	Als de visie bewerkbaar is, bevat de kolom READONLY de waarde Y. Indien de visie niet bewerkbaar is, bevat de kolom de waarde N.
JOIN	Char(1)	De kolom JOIN bevat de waarde Y als de visie in een join kan worden gebruikt. Indien de visie niet in een join kan worden gebruikt (als de definitie van de visie een clausule GROUP BY bevat), bevat de kolom de waarde N.
SQLTEXT	Char(200)	De kolom SQLTEXT bevat de tekst van het commando CREATE VIEW waarmee de visie wordt gedefinieerd.

Naslagshandboek van dBASE IV

Foutmeldingen



i 1 2 3 4 5 6 7 A B

C D E F G H In

Foutmeldingen

A

Deze bijlage bevat alle foutmeldingen van dBASE IV en SQL. De foutmeldingen voor beide gedeelten worden in alfabetische volgorde weergegeven. Het nummer van elke foutmelding wordt tussen vierkante haken achter de melding weergegeven.

Foutmeldingen van dBASE IV

Foutmeldingen waarvan het nummer wordt weergegeven, kunnen worden onderschept met het commando ON ERROR, behalve wanneer u de fout oplost door een alternatief uit het foutkader te kiezen. Een foutbehandelingsroutine die bij ON ERROR is opgegeven, kan met de functie ERROR() het foutmeldingsnummer terugzenden of met CERROR() het nummer van de laatste compileerfout terugzenden. Foutmeldingen die vanuit CERROR() kunnen worden teruggezonden, worden met een asterisk (*) weergegeven.

Meldingen zonder nummer worden door het Control Center gegenereerd en kunnen niet worden onderschept.

Wanneer dat mogelijk is, worden suggesties gedaan hoe u de oorzaak van de foutmelding kunt wegnemen.

Aan dit type zone kan geen groepzone worden toegevoegd.
Deze melding verschijnt wanneer u hebt geprobeerd een groepzone toe te voegen aan een zone die geen pagina-aanhef, rapportintro of groepintro is.

ACOS(): buiten bereik **[293]**
De waarde van de uitdrukking die u bij de arc-cosinusfunctie hebt gebruikt, valt buiten het bereik van -1,0 tot +1,0.

Afdrukcommando's kunnen niet worden genest **[337]**
Elk PRINTJOB-commando moet worden afgesloten met het commando END PRINTJOB voordat u een ander afdrukcommando kunt starten. Door het commando END PRINTJOB op te geven, wordt deze foutmelding verwijderd.

ALIAS niet gevonden:<aliasnaam> **[13]**
U hebt geprobeerd met SELECT een database-gebied te selecteren dat buiten het bereik van A tot en met J of 1 tot 10 valt, of u hebt een ongedefinieerde alias gebruikt.

ALIAS-naam is al in gebruik **[24]**
U hebt geprobeerd met USE een database-bestand in gebruik te nemen dat al is geopend, dat dezelfde alias heeft of dat een naam of alias heeft die binnen het standaardbereik van A tot en met J valt.

ALIAS-uitdrukking buiten bereik **[232]**
U hebt een aliasuitdrukking opgegeven die buiten het bereik van 1 tot 10 valt.

Alle database-bestanden moeten zijn gesloten voordat u PROTECT start **[173]**
Sluit alle databases voordat u PROTECT kiest.

Alleen de afmetingen van vensters, kaders en velden kunnen worden gewijzigd
Dit is een foutmelding van het Control Center. Deze melding verschijnt als u op **Shift-F7** drukt voor een object waarvan u de grootte niet kunt wijzigen.

Alleen leesrecht voor bestand **[111]**
U hebt geprobeerd naar een bestand weg te schrijven waarvoor u alleen leesrecht hebt of u hebt geprobeerd een bijbehorend .mdx-bestand te maken voor SQL-systeemcatalogusbestanden. Voor SQL-systeemcatalogusbestanden geldt leesrecht en daarom kan de .dbf-bestandsaanhef niet worden bijgewerkt met de aanduiding van een bestaand bijbehorend .mdx-bestand. U kunt echter andere typen indexbestanden maken (.ndx).

ALTERNATE<bestandsnaam> kan niet worden geopend **[72]**
U hebt een ongeldige ALTERNATE-bestandsnaam in het bestand Config.db opgegeven. Wijzig Config.db en de toekenning van de ALTERNATE-bestandsnaam.

ASIN(): buiten bereik **[291]**
De waarde van de uitdrukking die u bij de arc-sinusfunctie hebt opgegeven, valt buiten het bereik van -1,0 tot +1,0.

ATAN(): buiten bereik **[294]**
U hebt voor de functie ATAN() een te grote of te kleine numerieke uitdrukking opgegeven.

Basisrapport niet beschikbaar tijdens wijziging van formulier, rapport of etiket **[351]**
Tijdens het wijzigen van een formulier, rapport of een etiket kunt u de formulieroptie Basisrapport niet selecteren.

Begin van bestand bereikt **[38]**
U hebt geprobeerd in een database-bestand een zoekbewerking in achterwaartse richting uit te voeren maar de cursor bevond zich reeds aan het begin van de database.

Bestand bestaat al **[7]**
U hebt geprobeerd een bestand te maken of een nieuwe naam te geven maar de bestandsnaam komt al voor. Kies een andere naam.

Bestand bestaat niet:<bestandsnaam> **[1]**
Het opgegeven bestand werd niet aangetroffen. Controleer de bestandsnaam en het actieve pad.

Bestand in gebruik door<gebruikersnaam> **[372]**
In een multi-user omgeving is het bestand dat u wilde openen, vergrendeld door de gebruiker die in de foutmelding wordt genoemd.

Bestand in gebruik door een andere gebruiker **[108]**
Het bestand dat u wilde openen is met USE in gebruik genomen door een andere gebruiker in het netwerk. Probeer het later opnieuw.

Bestand is al geopend **[3]**
Het bestand dat u hebt geprobeerd te openen, is al geopend. Als u het commando USE voor het openen van een database-bestand gebruikt, is het bestand wellicht in een ander werkgebied geopend.

Bestand is al in catalogus opgenomen **[286]**
Deze melding verschijnt in het Control Center als u hebt geprobeerd een bestand aan de catalogus toe te voegen en de bestandsnaam zich al in de catalogus bevond.

Bestand is gewist **[49]**
U hebt geprobeerd een bestand te openen of te verwijderen dat al is gewist.

Bestand kan niet worden geladen
Deze melding verschijnt wanneer u hebt geprobeerd een bestand te laden dat niet overeenkomt met een van de indelingen (rapport-, formulier- en etiketbestanden) die door dBASE worden verwerkt. Deze melding kan ook verschijnen als het bestand is beschadigd.

Bestand kan niet worden gewijzigd omdat de ontwerpmodus is uitgeschakeld **[314]**
Deze melding wordt meestal bij toepassingsprogramma's weergegeven waarin het u niet is toegestaan de ontwerpmodus van dBASE IV te activeren. Als SET DESIGN op OFF staat, kunt u vanuit het Control Center of met dBASE IV-commando's geen bestanden maken of wijzigen, en kan evenmin de toetscombinatie **Shift-F2** voor EDIT, APPEND en BROWSE worden gebruikt. Stel SET DESIGN in op ON als u een bestand moet maken of wijzigen. Raadpleeg SET DESIGN.

Bestand moet in EXCLUSIVE-mode worden geopend **[110]**
U hebt geprobeerd een commando zoals PACK te geven voor een bestand dat u niet voor exclusief gebruik hebt geopend. Voordat u het commando weer kunt geven, sluit u het bestand en opent u het opnieuw terwijl u SET EXCLUSIVE op ON instelt.

Bestand niet aanwezig in transaction-log **[194]**
U kunt het commando ROLLBACK niet op een bestand uitvoeren waarvoor geen log-bestand van de transactie is vastgelegd. Dit bestand kunt u alleen met behulp van de reservekopieën terugbrengen naar de toestand waarin het zich bevond voor de transactie.

Bestand niet gekoppeld **[326]**
Tijdens een semantische controle is gebleken dat de bestanden in de definitie van de query niet zijn gekoppeld (query-ontwerp).

Bestand niet toegankelijk **[29]**
U hebt geprobeerd een bestand te openen waarvoor u alleen leesrecht hebt of waarvoor het besturingssysteem (DOS) een andere beperking op de toegang heeft opgelegd.

Bestandscatalogus is leeg
Er bevinden zich geen bestanden in de catalogus die u hebt geprobeerd te openen. Open een andere catalogus of maak een catalogus en neem daarin de gewenste bestanden op.

Bestandsstructuur kan niet verder worden uitgebreid
U hebt het onderste gedeelte van het gebied van de bestandsstructuur bereikt.

Bewerking niet toegestaan in rekenvelden **[371]**
U kunt het commando SORT of INDEX niet voor rekenvelden gebruiken.

Bewerking niet toegestaan in de rekenveldstructuur
U hebt geprobeerd gegevens in te voeren in het veld onder de bestandsnaam van een rekenveldstructuur.

Bewerking niet toegestaan met door de gebruiker gedefinieerde functie **[150]**
Bepaalde bewerkingen die niet zijn toegestaan met door de gebruiker gedefinieerde functies, worden beschreven in hoofdstuk 1, "Basisbegrippen" en bij SET DBTRAP in hoofdstuk 3.

Bijbehorend MDX-bestand is beschadigd **[289]**
Het bijbehorende indexbestand (.mdx) komt niet overeen met het database-bestand. Deze fout corrigeert u door voor het database-bestand een nieuwe index te maken. Op het scherm verschijnt de vraag of u wilt doorgaan.

Bijbehorend MDX-bestand niet gevonden:<bestandsnaam> **[210]**
Het bij het actieve database-bestand behorende .mdx-bestand is niet aangetroffen. Op het scherm verschijnt de vraag of u wilt doorgaan.

Bijwerken niet mogelijk zolang een fout is opgetreden **[395]**
In QBE hebt u **Bijwerken** gekozen om een mutatie uit te voeren, maar er komt een fout voor in de query-definitie.

Bron komt niet overeen met object **[95]**
Deze foutmelding verschijnt als er een programma wordt weergegeven en SET ECHO op ON is ingesteld, of als er code in het codevenster van de debugger wordt weergegeven en de broncode niet met de objectcode overeenkomt. Dit kan gebeuren wanneer een .prg-bestand is gewijzigd en niet opnieuw is gecompileerd. Onder deze omstandigheden komt de weergegeven broncode niet overeen met de uitgevoerde objectcode.

Bron-database moet zich in een niet-geselecteerd werkgebied bevinden **[189]**
U kunt met UPDATE alleen gegevens uit een database bijwerken als de database zich niet in het geselecteerde werkgebied bevindt.

Bronbestand<bestandsnaam> kan niet worden geopend **[398]**
Het opgegeven bronbestand (.res) kan niet worden gelezen. Deze ernstige fout leidt ertoe dat dBASE IV wordt beëindigd. U kunt nagaan waar het betreffende .res-bestand zich op de programmadiskettes bevindt en dit van de diskette naar de bestandsindex op de vaste schijf van waaruit u dBASE IV start, kopiëren. Als u het bestand niet aantreft, neemt u contact op met Ashton-Tate Customer Services voor hulp, want dBASE IV kan niet zonder dit bestand worden uitgevoerd.

CHANGE(), onvoldoende geheugen **[161]**
Deze foutmelding verschijnt als er onvoldoende geheugenruimte voor het uitvoeren van de functie CHANGE() is.

CHANGE(), record is door een ander vergrendeld **[160]**
Deze foutmelding verschijnt als u met de functie CHANGE() probeert te bepalen of het record is gewijzigd terwijl een andere gebruiker op dat moment het record heeft vergrendeld.

CHR(): buiten bereik **[57]**
Het argument dat u voor de functie CHR() hebt opgegeven, is kleiner dan 0 of groter dan 255. Dit is het bereik van de uitgebreide ASCII-tekenset van IBM.

COMMAND.COM kan niet worden geladen **[92]**
U hebt geprobeerd vanuit het Control Center naar DOS te gaan, maar hiervoor is onvoldoende RAM-geheugen beschikbaar. Sluit een aantal bestanden of verwijder geheugenvariabelen en probeer het opnieuw.

Commando alleen geldig in programma's **[260]***
U hebt geprobeerd een commando te geven dat alleen vanaf de commandostip in programma's is toegestaan. Schrijf een programma of een procedure zodat u dit commando kunt gebruiken.

Commando is onbereikbaar **[369]**
In het programma is een commando aanwezig dat niet kan worden uitgevoerd omdat RETURN of EXIT eraan voorafgaat. Commando's na het einde van een programma kunnen niet worden ingelezen of uitgevoerd.

Commando kan niet recursief worden aangeroepen **[103]**
Een LIST- of DISPLAY-commando kan niet recursief worden aangeroepen door een ander LIST- of DISPLAY-commando. U kunt bijvoorbeeld LIST niet met een UDF (door de gebruiker gedefinieerde functie) gebruiken als de UDF een ander LIST- of DISPLAY-commando aanroept. Een BROWSE-commando kan geen ander BROWSE- (of APPEND-, CHANGE-, EDIT- of INSERT-)commando in hetzelfde werkgebied aanroepen. Raadpleeg SET DBTRAP in hoofdstuk 3 voor meer informatie over recursieve commando's.

Commando niet bruikbaar in dBASE IV **[93]**
U hebt een commando zoals SET MENU of SET DOHISTORY gegeven dat in dBASE IV geen functie heeft, maar dat nodig is voor de compatibiliteit met eerdere versies van dBASE.

Commando niet toegestaan in deze context **[389]***
Dit commando kan niet in een door de gebruiker gedefinieerde functie of met ON COMMAND worden gebruikt.

Commando niet toegestaan in een door de gebruiker gedefinieerde functie **[384]***
Bepaalde commando's zijn, afhankelijk van de instelling van SET DBTRAP, niet toegestaan in UDF. Raadpleeg SET DBTRAP in hoofdstuk 3 voor meer informatie.

Commando niet toegestaan in programma's **[306]***
U hebt in een programmabestand een commando gegeven dat in programma's niet is toegestaan (zoals RESUME).

Commando niet toegestaan in SQL **[307]***
Dit dBASE-commando is in de SQL-modus niet toegestaan.

Commando onbekend **[16]***
U hebt een commando opgegeven dat geen deel uitmaakt van de syntaxis van dBASE IV.

Commando READ of @ ongeldig na afsluitende commando's in FMT-bestand **[359]***
Na het afsluitende gedeelte van een indelingsbestand (.fmt) is het commando READ of @ aangetroffen. Wanneer een indelingsbestand is beëindigd, zijn deze commando's niet toegestaan.

Commando te lang, open een venster met CTRL-HOME **[402]**
Op de commandoregel kunt u een commando van maximaal 254 tekens bewerken. Als u een langere commandoregel nodig hebt, drukt u op Ctrl-Home zodat u het commando in een venster kunt bewerken.

Commando USE niet geldig voor SQL-systeemtabellen **[218]**
U hebt geprobeerd met het dBASE-commando USE een SQL-systeemtabel te activeren. Het commando USE kan niet voor SQL-tabellen worden gebruikt.

Commando wordt niet ondersteund door RunTime **[290]**
Bepaalde commando's, zoals MODIFY STRUCTURE en CREATE SCREEN, zijn in RunTime-bestanden niet toegestaan. Het commandobestand dat u hebt gemaakt, bevat een van deze commando's.

Compileerfout **[360]**
Het programmabestand is vanwege syntaxisfouten niet gecompileerd. Corrigeer deze fouten in het programmabestand en compileer het opnieuw.

CONTINUE zonder LOCATE **[42]**
U kunt het commando CONTINUE niet zonder het commando LOCATE uitvoeren.

Coördinaten in indelingsbestand moeten in oplopende volgorde staan
De schermcoördinaten voor de @-commando's moeten in oplopende volgorde staan voordat u dit bestand naar PFS:FILE kunt exporteren.

Coördinaten liggen buiten het scherm **[30]**
Bij het definiëren of verplaatsen van een venster hebt u rijnummers en kolomnummers opgegeven die zich buiten het scherm bevinden. Bij 25 regels bevinden de grenzen van het scherm zich op 0,0 en 24,79.

Coördinaten van venster liggen buiten het scherm **[332]**
U hebt geprobeerd een venster te definiëren dat zich niet binnen de coördinaten 0,0 en 22,79 bevindt. Voor DISPLAY=EGA43 gelden de coördinaten 0,0 en 38,79.

Cursor kan niet voorbij onderkant (linkerkant, rechterkant of bovenkant) van venster worden verplaatst
U hebt geprobeerd de cursor buiten het ontwerpscherm te plaatsen.

Cyclische relatie **[44]**
U hebt een oneindige lus gemaakt door in de actieve geselecteerde database een relatie te leggen die naar zichzelf verwijst.

Database is gecodeerd **[131]**
U hebt geprobeerd een gecodeerde database te openen zonder via PROTECT en het aanmeldingsscherm uw wachtwoord in te voeren.

Database niet geïndexeerd **[26]**
Voordat u dit commando kunt geven, moet u een index maken.

DBT-bestand kan niet worden geopend **[41]**
Het memobestand kan niet worden geopend omdat het is vergrendeld of niet beschikbaar is.

De query is niet uitvoerbaar, maar is toch opgeslagen
U hebt een onjuist gedefinieerde query opgeslagen.

De totale etiketbreedte overschrijdt het maximum
Deze melding verschijnt tijdens het genereren van etiketten uit dBASE III PLUS.

Een etiket mag maximaal 250 kolommen breed zijn.

Debugger is al geactiveerd **[347]**
U hebt geprobeerd de debugger aan te roepen terwijl deze al in gebruik is.

Deze gebruikersnaam is niet gedefinieerd **[156]**
In het hulpprogramma PROTECT is de gewenste naam niet aanwezig. Informeer bij de systeembeheerder.

Deze groepnaam is niet gedefinieerd **[155]**
U moet een groepnaam definiëren voordat u met het commando PROTECT een bestand kunt coderen.

Deze weergavemodus is niet beschikbaar **[216]**
De geselecteerde weergavemodus is niet beschikbaar. Gebruik het schermgrote menu van het SET-commando, zodat u de ondersteunde selecties kunt bekijken.

Diskette is voorzien van schrijfbeveiliging
U hebt geprobeerd een bestand weg te schrijven naar een diskette met schrijfbeveiliging. Verwijder het schrijfbeveiligingsplakkertje.

Dit bestand kan niet worden gewijzigd **[342]**
Deze melding duidt aan dat het bestand waarop de cursor zich in het Control Center bevindt, niet kan worden gewijzigd wanneer op de ontwerptoets **(Shift-F2)** wordt gedrukt. Bestanden met de bestandsnaamextensies .qbo, .upo, .fmo, .fro, .lbo, .dbo, .exe en .com kunnen niet worden gewijzigd.

Dit is geen MDX-indexbestand **[206]**
Het bestand dat u hebt geprobeerd te openen, is geen geldig .mdx-bestand.

Dit wachtwoord is niet gedefinieerd **[157]**
Voor het commando PROTECT hebt u geen wachtwoord. Raadpleeg de systeembeheerder.

Doelbestand niet gevonden in catalogus **[343]**
Als u vanuit het Control Center een mutatie-query uitvoert en het .upd-bestand niet in de catalogus wordt aangetroffen, verschijnt deze foutmelding.

Door gebruiker gedefinieerde functies moeten in een waarde resulteren **[382]***
Het resultaat van alle door de gebruiker gedefinieerde functies moet een waarde zijn. Wijzig de door de gebruiker gedefinieerde functie zodanig dat deze resulteert in een waarde.

Door SQL gemaakte TAG kan niet worden gewist:<naam TAG> **[376]**
U kunt een TAG die met SQL is gemaakt, alleen tijdens de interactieve SQL-modus verwijderen.

Door SQL gemaakte TAG kan niet worden overschreven:<naam TAG> **[377]**
Bijbehorende .mdx-bestanden kunnen labels bevatten die met SQL of dBASE zijn gemaakt. Deze foutmelding geeft aan dat u met dBASE een indexlabel hebt gemaakt die dezelfde naam heeft als een bestaande SQL-indexlabel.

Een bestand dat tegen overschrijven is beveiligd, kan niet worden gewist **[336]**
Een bestand waarvoor u alleen leesrecht hebt, kunt u niet wissen. De systeembeheerder kan u informeren over uw toegangsrechten.

Een bestandsstructuur kan niet aan zichzelf worden gekoppeld **[345]**
U hebt geprobeerd twee velden in dezelfde bestandsstructuur te koppelen.

Een leeg veld kan niet worden verplaatst
Dit is een foutmelding van het database-ontwerp van het Control Center. Deze melding verschijnt wanneer de cursor zich in een leeg veld in de sorteertabel bevindt en u op **F7** drukt.

Een menu dat in gebruik is, kan niet worden gewijzigd **[174]**
U hebt geprobeerd het commando DEFINE MENU voor het actieve menu te geven. Geef eerst het commando DEACTIVATE MENU en dan DEFINE MENU.

Een menu dat in gebruik is, kan niet worden gewist **[176]**
Een actief menu op het scherm kan niet met het commando CLEAR MENU of RELEASE MENU worden gewist.

Een pop-up die in gebruik is, kan niet worden gewijzigd **[175]**
U hebt geprobeerd het commando DEFINE POPUP voor de actieve pop-up te geven. Geef eerst het commando DEACTIVATE POPUP en dan DEFINE POPUP.

Een popup die in gebruik is, kan niet worden gewist **[177]**
Een actief pop-up menu op het scherm kan niet met het commando CLEAR POPUP of RELEASE POPUP worden gewist.

Een query kan slechts één unieke verzamelfunctie bevatten **[324]**
U mag bij een verzamelfunctie slechts één UNIQUE-opdracht geven.

Een REPLACE-query moet ten minste één WITH-operator bevatten **[370]**
U hebt geprobeerd een mutatie-query (vervangen) uit te voeren zonder WITH-opdracht.

Einde van bestand bereikt **[4]***
U hebt geprobeerd in een database-bestand een zoekbewerking in voorwaartse richting uit te voeren maar de cursor bevond zich reeds aan het einde van de database, of u hebt met SKIP het einde van het bestand overgeslagen.

END PRINTJOB voor vorige PRINTJOB ontbreekt **[300]***
U hebt geprobeerd het PRINTJOB-commando uit te voeren terwijl zich in de wachtrij nog een afdrukcommando bevond of terwijl er nog een afdrukcommando werd uitgevoerd. Geef het commando END PRINTJOB voordat u een nieuw afdrukcommando start. Afdrukcommando's kunnen niet worden genest.

END TRANSACTION voor vorige BEGIN TRANSACTION ontbreekt **[301]***
Elk BEGIN TRANSACTION-commando moet worden afgesloten met een END TRANSACTION-commando. Transacties kunnen niet worden genest.

ENDCASE voor vorig commando DO CASE ontbreekt **[247]***
Elk DO CASE-commando moet worden afgesloten met een ENDCASE-commando.

ENDDO voor vorig commando DO WHILE ontbreekt **[246]***
Elk DO WHILE-commando moet worden afgesloten met een ENDDO-commando.

ENDIF voor vorig commando IF ontbreekt **[244]***
Elke IF-opdracht moet worden afgesloten met een ENDIF-opdracht.

ENDIF voor vorig commando IF/ELSE ontbreekt **[245]***
Elke IF/ELSE-voorwaarde moet worden afgesloten met een ENDIF-opdracht.

ENDSCAN voor vorig commando SCAN ontbreekt **[269]***
Een SCAN-commando moet worden afgesloten met ENDSCAN.

ENDTEXT voor vorige TEXT ontbreekt **[341]***
Deze foutmelding verschijnt wanneer er zich in een procedure na een TEXT-commando geen ENDTEXT-commando bevindt.

Er is geen afdruk in uitvoering **[282]**
U hebt geprobeerd het commando END PRINTJOB te geven terwijl er geen actief commando PRINTJOB was dat kon worden beëindigd.

Er is geen catalogus gemaakt **[122]**
U moet een catalogus eerst met CREATE maken voordat u deze kunt openen.

Er is geen groepnaam vastgesteld **[158]**
U moet een groepnaam vaststellen voordat u met het commando PROTECT een bestand kunt coderen.

Er kan slechts één MDX-bestand per database geopend zijn **[211]**
U hebt geprobeerd een tweede bijbehorend indexbestand (.mdx) te openen.

EVERY alleen toegestaan met één paar voorbeeldvariabelen **[318]**
Dit is een QBE-foutmelding die aanduidt dat een volledige join van de buitenste elementen niet is toegestaan (outer join).

Extra tekens aan het einde van commando genegeerd **[151]**
Er bevinden zich aan het einde van de opdracht extra tekens. Deze tekens zijn overbodig en worden door het programma genegeerd.

FIRST alleen toegestaan bij één helft van een paar voorbeeldvariabelen
In QBE hebt u FIRST vóór de beide helften van een voorbeeldvariabele opgegeven, bijvoorbeeld FIRST x in een bestandsstructuur en FIRST x in een andere bestandsstructuur.

Fout bij uitvoering van **[80]**
U hebt een functie van een onjuist argument voorzien. Achter deze melding wordt de naam van de functie die u wilde uitvoeren, weergegeven. Voorbeeld:

*****Fout bij uitvoering van SUBSTR(): beginpunt buiten bereik**

Fout in configuratiewaarde **[145]**
In het bestand Config.db bevindt zich een ongeldige waarde. Raadpleeg *Aan de slag met dBASE IV* voor een bespreking van de configuratieparameters in het bestand Config.db.

Fout op regel<nummer> **[96]**
Dit is een melding van de compiler die het regelnummer bevat waarin de syntaxisfout is opgetreden.

Fout tijdens lezen van log-bestand **[192]**
Het log-bestand van een transactie is beschadigd of kan om een andere reden niet worden gelezen. Het commando ROLLBACK kan niet worden uitgevoerd. Raadpleeg RESET in hoofdstuk 2.

Fout tijdens verwerking van commando ON ERROR **[394]**
Er is een fout opgetreden in het commando of het programma dat tijdens een uitzonderingsverwerking (ON ERROR) is uitgevoerd. Met andere woorden: tijdens de uitvoering van het<commando> in ON ERROR<commando> is een fout opgetreden.

Fouten in definitie van query - niet uitvoerbaar **[352]**
In de definitie van de query bevinden zich fouten. Controleer de syntaxis en probeer de query opnieuw uit te voeren.

Fouten in definitie van query - niet uitvoerbaar
U hebt in het Control Center een query-bestand dat fouten bevatte, gemarkeerd en op **F2** gedrukt.

Functie niet beschikbaar in SQL-modus:<functienaam> **[340]***
De functie die in de foutmelding wordt genoemd, kan niet in de SQL-modus worden gebruikt.

Functie niet gevonden:<functienaam> **[368]**
De door de gebruiker gedefinieerde functie kan de bijbehorende procedure FUNCTION niet vinden in de geopende programma- en procedurebestanden.

Geen array **[229]**
U hebt geprobeerd een geheugenvariabele als reeks te gebruiken maar deze variabele was niet als reeks gedefinieerd.

Geen BEGIN TRANSACTION gegeven die overeenkomt met dit commando **[303]***
U hebt het commando END TRANSACTION ingevoerd maar er is geen actief commando BEGIN TRANSACTION dat kan worden beëindigd.

Geen bestanden van gevraagd type aanwezig. Druk op een toets om door te gaan
U hebt uit het schermgrote menu **SET** de optie **Indeling** of **Index** geselecteerd maar in de actieve bestandsindex bevonden zich geen bestanden met de gewenste extensie.

Geen bestanden van gevraagd type in catalogus
U hebt uit het schermgrote menu **SET** de optie **Indeling** of **Index** geselecteerd maar in de actieve catalogus bevonden zich geen bestanden met de gewenste bestandsnaamextensie die in de bestandsnamenlijst konden worden weergegeven.

Geen bestanden van gevraagd type in catalogus. Druk op een toets om door te gaan
De geopende catalogus bevat geen bestanden van het gezochte type.

Geen database geopend **[52]**
U hebt een commando opgegeven waarvoor een geopend database-bestand nodig is. Open met het commando USE een database-bestand.

Geen database geopend. Formulier bevat de volgende velden:
Deze foutmelding wordt in het menu **Formulier** weergegeven wanneer het formulier dat wordt gewijzigd, naar een database verwijst die niet geopend is. De lijst met velden die door dit formulier worden gebruikt, wordt ook weergegeven.

Geen database geopend. Voer een bestandsnaam in:
U hebt geprobeerd een object te maken maar er is geen database-bestand of visie actief.

Geen dBASE-database **[15]**
U hebt geprobeerd in dBASE IV een databasebestand te gebruiken dat niet de juiste dBASE IV-indeling heeft. Deze melding kan ook betekenen dat u hebt geprobeerd een database-bestand met een dBASE II-indeling te importeren zonder eerst de extensie te wijzigen in .db2.

Geen DBO-bestand:<bestandsnaam> **[253]**
De bestandsnaam die u achter het commando DO hebt ingevoerd betreft geen gecompileerd programmabestand, ook al heeft de naam een .dbo-extensie. Zoek het gewenste bestand in de bestandsindex. Compileer het .prg-bestand opnieuw.

Geen DO CASE gegeven die overeenkomt met dit commando **[250]***
Aan elk commando ENDCASE moet het commando DO CASE voorafgaan. Controleer het programma en corrigeer de onderdelen van de syntaxis die niet met elkaar in balans zijn.

Geen DO WHILE gegeven die overeenkomt met dit commando **[251]***
Aan elk commando ENDDO moet het commando DO WHILE voorafgaan.

Geen DO WHILE/SCAN/PRINTJOB die overeenkomt met dit commando **[304]***
Aan het commando ENDDO/ENDSCAN/END PRINTJOB gaat geen commando DO/SCAN/PRINTJOB vooraf dat nog moet worden uitgevoerd.

Geen geldig PFS-bestand **[140]**
Het bestand dat u hebt geprobeerd te importeren, is geen geldig PFS:FILE-bestand.

Geen geldig query-bestand **[134]**
U hebt geprobeerd met SET FILTER TO een filter voor een ongeldig query-bestand in te stellen. Maak een nieuw query-bestand.

Geen geldig vensterbestand **[358]**
Het bestand met vensterdefinities dat u hebt geprobeerd te activeren, is niet als vensterbestand benoemd. Controleer de bestandsnaam en probeer het opnieuw.

Geen geldig visiebestand **[127]**
U hebt geprobeerd met SET VIEW TO of MODIFY VIEW een visie in te stellen of te wijzigen voor een bestand dat mogelijk is beschadigd. Voer de commando's CREATE VIEW FROM ENVIRONMENT of CREATE VIEW nogmaals uit, zodat er een nieuw visiebestand wordt gemaakt.

Geen geldige dBASE II-database **[257]**
De database die u hebt geprobeerd te importeren, is geen geldige dBASE II-database of u hebt de extensie niet in .db2 gewijzigd.

Geen geldige Framework II-database/spreadsheet **[256]**
Het bestand dat u hebt geprobeerd te importeren, is geen Framework II-bestand.

Geen geldige RapidFile-database **[255]**
Het bestand dat u hebt geprobeerd te importeren, is geen geldig RapidFile-bestand.

Geen geldige uitdrukking **[233]**
U hebt in een programma een ongeldige dBASE IV-uitdrukking opgegeven. Zoek de juiste syntaxis in dit naslaghandboek op.

Geen geldige uitdrukking. Druk op een toets om door te gaan
Het veld heeft een veldnaam die niet geldig is in dBASE IV. Deze foutmelding verschijnt ook wanneer u een onbenoemd rekenveld aan de visie probeert toe te voegen.

Geen IF gegeven die overeenkomt met dit commando **[249]***
Aan elk commando ENDIF moet een IF-opdracht voorafgaan.

Geen kopieerbare velden gevonden **[138]**
U hebt geprobeerd velden te kopiëren maar u hebt niet eerst een veldenlijst gedefinieerd.

Geen logische uitdrukking **[37]**
Voor dit programma of deze procedure (bijvoorbeeld in een DO/WHILE-clausule) is logisch waar (.T.) of onwaar (.F.) vereist.

Geen numerieke uitdrukking **[27]**
U hebt geprobeerd bij een rekenkundige, statistische of trigonometrische functie of bij een commando waarvoor een numerieke uitdrukking als argument is vereist, een niet-numerieke uitdrukking op te geven.

Geen PRINTJOB gegeven die overeenkomt met dit commando **[305]***
U hebt het commando END PRINTJOB gegeven maar er is geen commando PRINTJOB dat moet worden beëindigd.

Geen programmabestand gevonden voor deze toepassing **[334]**
Het .prg-bestand voor de toepassing die u wilde uitvoeren, bevindt zich niet in de actieve bestandsindex of op het zoekpad. Zoek het programmabestand op schijf op en gebruik dan de juiste bestandsspecificatie.

Geen records geselecteerd **[365]**
Deze melding verschijnt in BROWSE en duidt aan dat er geen records zijn waarin u kunt bladeren. De fout kan optreden wanneer er een filter (SET FILTER TO) wordt gebruikt en er voor BROWSE geen records zijn geselecteerd waarin kan worden gebladerd.

Geen SCAN gegeven die overeenkomt met dit commando **[302]***
U hebt het commando ENDSCAN gegeven maar er is geen commando SCAN dat moet worden beëindigd.

Geen selectie gemaakt voor MOVE
Selecteer het te verplaatsen veld door op **F6** te drukken voordat u op **F7** drukt.

Geen tekstuitdrukking **[45]**
U hebt de voor een veld vereiste tekenuitdrukking niet opgegeven (u hebt bijvoorbeeld geprobeerd bij een numerieke uitdrukking scheidingstekens aan te brengen).

Geen toegangsrecht op dit niveau **[133]**
U hebt geprobeerd een bestand op te roepen waarvoor u geen toegangsrecht hebt. Informeer bij de systeembeheerder naar toegangsniveaus.

Geen velden beschikbaar **[47]**
U hebt geprobeerd bij een database-bestand waarvan zich geen velden in de SET FIELDS-lijst bevinden, een commando zoals BROWSE of EDIT te geven.

Geen velden van het gevraagde type aanwezig. Druk op een toets om door te gaan
U hebt in het menu **Velden** om een veld verzocht terwijl er zich in de bestandsstructuur of in de SET FIELDS-lijst geen velden van het gewenste type bevonden.

Geen zoekreeks opgegeven
U hebt in het menu **Ga naar** de optie **Voorwaarts/Achterwaarts zoeken** geselecteerd of op **Shift-F5** gedrukt voordat u een zoekreeks had ingevoerd.

Gegevensfout
Er bevindt zich op de vaste schijf of op de diskette een onbruikbare sector die dBASE IV heeft geprobeerd te lezen. Deze ernstige fout kan gegevensverlies veroorzaken.

Gegevenstype onjuist **[9]***
U hebt geprobeerd in één bewerking verschillende gegevenstypen te combineren. Bewerkingen die deze melding tot gevolg hebben, zijn: een uitdrukking die verschillende gegevenstypen bevat, een REPLACE-commando met een ander gegevenstype dan het veld, een SEEK-commando dat niet overeenkomt met het gegevenstype van de indexsleutel of een SORT-commando dat u op een logisch veld of memoveld probeert uit te voeren.

Geheugenfout
Dit is een foutmelding van DOS die aanduidt dat het RAM-geheugen beschadigd is. Verlaat dBASE IV en voer het bij de computer geleverde diagnoseprogramma uit. U kunt hiermee de lokatie van de defecte RAM-chip(s) bepalen. Als u niet deskundig bent op het gebied van computerapparatuur, waarschuwt u een technicus voor de diagnose en het vervangen van de defecte geheugenchips.

Geheugenvariabele kan hier niet als PRIVATE worden gedefinieerd **[270]**
U hebt geprobeerd een PRIVATE-commando te geven voor een geheugenvariabele die al op het huidige niveau is gedefinieerd.

Geopend bestand kan niet worden gewist **[89]**
Voordat u database-bestanden of indexbestanden met ERASE kunt wissen, moet u deze eerst sluiten.

Gevraagde database kan niet worden geselecteerd **[17]**
U hebt een ongeldig nummer voor een werkgebied gekozen. Geef een getal op tussen 1 en 10 of een letter tussen A en J.

GROUP BY gebruikt zonder verzameloperator [327]
Tijdens een semantische controle heeft dBASE IV de operator "GROUP BY" aangetroffen maar geen corresponderende verzameloperator (query-ontwerp).

GROUP BY kan slechts eenmaal in een kolom voorkomen [339]
U hebt GROUP BY in één kolom tweemaal ingevoerd.

Helft van paar voorbeeldvariabelen ontbreekt [363]
U hebt een voorbeeldvariabele in een QBE-bestandsstructuur ingevoerd zonder de benodigde bijbehorende voorbeeldvariabele in een andere bestandsstructuur in te voeren. Als u bijvoorbeeld "WITH x" in een REPLACE-query invoert, moet u x in een andere bestandsstructuur definiëren. Controleer of beide helften van een voorbeeldvariabelenpaar juist zijn gespeld.

Het beschadigde indexbestand is onherstelbaar [364]
De inhoud van een indexbestand is onherstelbaar beschadigd. U maakt een nieuwe index voor het database-bestand door een nieuw INDEX-commando uit te voeren.

Het volgende station is niet gereed:<station>
De letter van het station staat aan het einde van deze melding. U kunt nog geen bestanden inlezen uit of wegschrijven naar het gewenste station. Sluit daarom eerst het deurtje van uw diskettestation.

Huidig printerstuurprogramma niet geschikt voor correspondentiekwaliteit [331]
Het geselecteerde printerstuurprogramma ondersteunt het afdrukken met correspondentiekwaliteit of met speciale letterfonts niet. Selecteer met de systeemgeheugenvariabele _pdriver="bestandsnaam.pr2" vanaf de commandostip het juiste printerstuurprogramma. Hierbij vervangt u "bestandsnaam" door de naam van het juiste printerstuurprogramma.

In een mutatie-query kunnen geen velden aan de visie worden toegevoegd
U hebt geprobeerd velden aan de visie toe te voegen maar in het veld onder de bestandsnaam van een bestandsstructuur bevond zich een mutatiecommando.

In een query-ontwerp kunnen niet meer dan acht bestanden worden geladen
In QBE (query by example) kunt u maximaal acht bestandsstructuren gebruiken.

In een veld dat een fout bevat, kan geen koppeling worden aangebracht [346]
U hebt geprobeerd een koppeling te leggen met een veld waarin zich een fout bevindt.

Index beschadigd. Stel index opnieuw samen voordat u gegevens raadpleegt [114]
Het indexbestand dat hoort bij de database die met USE in gebruik is genomen, is beschadigd. Maak met het commando REINDEX een nieuwe index.

Index is te groot (maximaal 100 tekens) [23]
Het resultaat van de indexsleutel waarop u met INDEX hebt geïndexeerd, is langer dan 100 tekens. Deze beperking heeft betrekking op de feitelijke lengte van de sleutel nadat de indexsleutel is geëvalueerd.

INDEX mislukt - sleutel is niet DISTINCT [236]
Kies een andere sleuteluitdrukking op basis waarvan u met INDEX ON een index maakt.

Index voor ARRAY-verwijzing ontbreekt [299]
Deze foutmelding verschijnt als u een reeksnaam zonder indexen opgeeft. Als u bijvoorbeeld een reeks xxx opgeeft en u een van de elementen op een waarde instelt en daarna probeert de waarde van de reeks weer te geven zonder de bijbehorende index op te geven, verschijnt deze foutmelding.

Indexbestand komt niet overeen met database **[19]**
Het indexbestand dat u met USE probeert te openen, bevat geen sleutelwoord dat is gerelateerd aan de database die met USE in gebruik is genomen. Als de database geen indexbestand heeft, maakt u een nieuw indexbestand.

Indexeren onderbroken. Eventuele onvolledige index wordt gewist **[113]**
U hebt tijdens de uitvoering van INDEX of REINDEX op Esc gedrukt. U kunt het indexeren voortzetten of annuleren door op de vraag van deze foutmelding met het gewenste antwoord te reageren.

Indexlabel bestaat al:<labelnaam> **[205]**
De indexlabel die u hebt opgegeven, bestaat al. Kies daarom een andere label.

Indexsleutel is te lang (maximaal 220 tekens) **[112]***
De sleuteluitdrukking op basis waarvan u met INDEX een index maakt, bestaat uit meer dan 220 tekens. Deze beperking heeft betrekking op het aantal tekens in de indexsleutel en niet op de feitelijke lengte van de resulterende sleutel.

Indexveld moet J of N zijn
Deze melding verschijnt wanneer u in de kolom Index van de ontwerpstructuur van de database op een ongeldige toets drukt.

Intern: Virtual Stack Overflow **[199]**
Deze foutmelding verschijnt als het stapelgeheugen voor de interne analyse overloopt. De fout kan optreden als een uitdrukking te gecompliceerd is.

Interne fout bij genereren van SQL-code **[311]***
Deze melding kan verschijnen als er zich tijdens het genereren van SQL-code een interne inconsistentie voordoet die problemen in dBASE IV veroorzaakt. Deze melding mag eigenlijk niet voorkomen maar als dit toch het geval is, neemt u contact op met Ashton-Tate Customer Services.

Interne fout in tekstverwerker. Bewerkbuffer kan zijn beschadigd
Dit is een ernstige fout die niet mag optreden. Wanneer deze melding verschijnt, neemt u contact op met Ashton-Tate Customer Services.

Interne fout: CMDSET(): **[66]**
Deze fout mag niet optreden. Deze interne foutmelding verschijnt wanneer een programmeur de interne SET-routine met een foutieve parameter oproept. Het .dbo-bestand is beschadigd en bevat token-code die als een ongeldige token-code van een SET-commando kan worden geïnterpreteerd.

Interne fout: Interne overloop van tabel **[69]**
Deze fout duidt aan dat dBASE geen ruimte meer heeft in de tabel voor de syntaxiscontrole van een uiterst gecompliceerde uitdrukking. De oplossing is de uitdrukking te vereenvoudigen.

Omvangrijke uitdrukkingen en/of velden kunnen een grotere syntaxiscontroletabel nodig hebben. Probeer de parameter EXPSIZE=<Nuitdr> in Config.db op 500 in te stellen in plaats van de minimum- en standaardwaarde 100.

Interne fout: Onbekende commandocode: **[65]**
Deze melding kan verschijnen als het gebruikte .dbo-bestand ongeldig is. Na de dubbele punt staat de decimale notatie van de ongeldige opcode.

Interne fout: Ongeldige opcode **[68]**
Dit is een interne fout. De compiler heeft een nummer voor een niet-bestaand commando gegenereerd.

Interne fout: Overloop van werkgebied EVAL **[67]**
De vaststelling hiervan gebeurt door een interne analyseroutine van dBASE die EVAL wordt genoemd. De tekenreeks die u vanaf de commandostip hebt ingevoerd, veroorzaakt een interne overloop van het stapelgeheugen. Controleer het aantal geneste functie-oproepen in de tekenreeks.

Invoegen niet mogelijk. Uitdrukking heeft maximale lengte
In een database-veld kunt u geen uitdrukking invoegen als de nieuwe uitdrukking de lengte van het veld overschrijdt.

JOIN kan niet met het bestand zelf **[139]**
U kunt alleen twee verschillende database-bestanden samenvoegen. Bij het commando USE moet u de naam van een andere database opgeven.

Kader selecteren met F6
Dit is een foutmelding van het Control Center. Deze melding verschijnt wanneer u tijdens een uitgebreide selectie van een kader hebt geprobeerd de stuurtoetsen te gebruiken.

Kaderrand niet gedefinieerd **[228]**
Deze foutmelding verschijnt wanneer u voor het afdrukken de randen van een kader definieert en daarbij de definitie van de randen weglaat of ongeldige definities voor de randen opgeeft.

Kolom is vol (maximum is 255 tekens)
Deze melding verschijnt wanneer u voor een kolom **Koppeling leggen door aanwijzen of Sorteren op dit veld** selecteert en deze kolom niet voldoende tekens over heeft voor de voorbeeldvariabele of de sorteerinstructies.

Kolomnummer moet tussen 0 en rechtermarge of 255 zijn **[223]**
U hebt een printer-kolomnummer opgegeven dat zich buiten de pagina bevindt. Geef het commando opnieuw en geef een kolomnummer op dat zich binnen de pagina bevindt.

Koppeling in voorbereiding
U hebt **Koppeling leggen door aanwijzen** geselecteerd en daarna hebt u deze optie opnieuw geselecteerd zonder de eerste koppeling, die nog moest worden uitgevoerd, af te sluiten (query-ontwerp).

Koppeling met memoveld of logisch veld niet mogelijk **[316]**
U hebt in QBE een voorbeeldvariabele in een memoveld of logisch veld ingevoerd.

Koppeling met tekenveld van meer dan 100 tekens niet mogelijk **[315]**
U hebt geprobeerd QBE-bestandsstructuren te koppelen met een tekenveld waarvoor een breedte van meer dan 100 tekens is opgegeven.

Leeg of onjuist veld kan niet aan visie worden toegevoegd
U hebt geprobeerd een ongeldig rekenveld aan de visie toe te voegen.

Leesfout **[254]**
Dit is een foutmelding van het besturingssysteem die een probleem met een leesbewerking aanduidt.

Leesfout in station<station>
Deze melding bevat de naam van het station waar de leesfout is opgetreden. Dit is een foutmelding van het besturingssysteem die een probleem met een leesbewerking aanduidt.

Lege structuur kan niet worden opgeslagen
Door op **Ctrl**-W of **Ctrl-End** te drukken, hebt u geprobeerd een lege database-structuur op te slaan. De menu-opties **Bestandsstructuur van database opslaan en Wijzigingen opslaan en stop** kunnen niet worden gekozen wanneer de structuur leeg is.

LET OP: bestandsdefinitie gewijzigd **[320]**
Nadat u de query de laatste maal hebt gewijzigd, is de structuur van één van de bestanden gewijzigd.

LET OP: bij wijzigingen in veld gaan CONVERT-gegevens verloren
Als u het veld _dbaselock in een database-bestand CONVERT verwijderd of wijzigt, kunt u CHANGE() en LKSYS() niet gebruiken.

Let op: indexsleutel bevat ALIAS of MEMVAR **[387]**
Door deze melding wordt u gewaarschuwd dat, wanneer de dBASE IV-omgeving wordt gewijzigd, een uitdrukking van een indexsleutel waarin ALIAS of een geheugenvariabele is opgegeven, ongeldig kan worden.

Linkermarge plus inspringing moet minder zijn dan rechtermarge **[221]**
Bij het opgeven van de printeruitvoer was de som van de waarden voor de linkermarge en de inspringing van de alinea groter dan de waarde voor de rechtermarge.

LOCK niet mogelijk **[129]**
Het record dat u wilde vergrendelen, is al door een andere gebruiker geactiveerd.

LOG(): nul of negatief **[58]**
De waarde van het argument van de functie voor het natuurlijke logaritme mag geen negatief getal of nul zijn.

Log-bestand is beschadigd **[195]**
Het log-bestand van een transactie dat door dBASE IV is bijgehouden, kan niet meer worden gelezen. Verwijder met het commando END TRANSACTION de bestandsaanhef van de actieve transactielabels. Zonder het log-bestand van een transactie kunt u ROLLBACK niet uitvoeren. Alleen met behulp van de reservekopie kunt u het bestand weer terugbrengen in de toestand van voor de transactie.

Log-bestand niet gevonden:<bestandsnaam> **[191]**
Het log-bestand van een transactie dat door dBASE IV is bijgehouden, is niet aangetroffen. Verwijder met het commando END TRANSACTION de bestandsaanhef van de actieve transactielabels.

Log-record komt niet overeen met database-record **[196]**
Het log-bestand van een transactie komt niet overeen met de database. Verwijder met het commando END TRANSACTION de bestandsaanhef van de actieve transactielabels.

LOG10(): nul of negatief **[292]**
De waarde van het argument voor een logaritme met grondtal 10 mag geen negatief getal of nul zijn.

Macro niet gevonden in de actieve bibliotheek **[355]**
De gebruikte macronaam of macrotoetscode is niet gedefinieerd in de macrobibliotheek die u hebt geladen. Als u een nieuwe macro maakt terwijl er zich in het geheugen een actieve bibliotheek bevindt, overschrijft u dubbele macronamen. Zorg ervoor dat u een unieke macronaam of toetscode definieert.

Macro wordt opgenomen [357]
Dit is geen foutmelding maar een mededeling die op de informatieregel verschijnt wanneer u een macro vastlegt.

Macro's te diep genest:<aanroepvolgorde> [361]
U hebt geprobeerd meer dan 16 macro's te nesten.

Maximaal één verzameloperator per kolom [380]
U hebt meer dan één verzamel-operator in een kolom opgegeven.

Maximale afdrukbreedte van 255 tekens overschreden
Deze foutmelding verschijnt wanneer de etiketbreedte vermenigvuldigd met het aantal etiketten meer dan 255 tekens bedraagt.

Maximale nesting voor compiler overschreden [248]*
In dBASE IV is per procedure maximaal 64Kb aan gecompileerde code toegestaan en in een lus van een sprongopdracht 32Kb aan gecompileerde code. Verklein de lussen en/of procedures van het programma.

Maximale recordlengte overschreden in WK1-bestand [392]
In het Lotus-werkblad is de maximum lengte van een record (4001 bytes) overschreden.

Maximum aantal GET-commando's overschreden [310]*
Verminder het aantal GET-commando's dat zich op één pagina bevindt.

Maximum aantal geheugenvariabelen bereikt [22]
Er is onvoldoende geheugenruimte in het systeem voor het definiëren van extra geheugenvariabelen.

Maximum aantal indexlabels is reeds bereikt
U hebt geprobeerd een achtenveertigste index aan het .mdx-bestand toe te voegen door in het database-ontwerp meer dan 47 indexvelden op J in te stellen.

Maximum aantal NDX-indexen is bereikt [204]
Er kunnen tien .ndx- of .mdx-bestanden tegelijk worden geopend. Sluit een aantal indexen voordat u probeert meer indexbestanden te openen.

Maximum aantal procedures overschreden [397]*
Het maximum aantal procedures voor één .dbo-bestand is overschreden.

Maximum aantal sorteerinstructies is 9 [309]
U hebt meer dan negen sorteerinstructies, bijvoorbeeld **ASC10**, in een QBE-bestandsstructuur ingevoerd.

Maximum aantal symbolen in compileerfase overschreden [274]*
Het begrip "symbolen in compileerfase" heeft betrekking op het totaal aantal unieke namen van geheugenvariabelen en database-velden in een gecompileerde procedure. Het standaardaantal compileerfasesymbolen is 500. Dit aantal is overschreden en moet dus worden verminderd.

Een andere oplossing is in het bestand Config.db dit aantal te verhogen door de waarde die aan CTMAXSYMS is toegewezen, te wijzigen. Raadpleeg *Aan de slag met dBASE IV*.

Maximum aantal symbolen tijdens uitvoering overschreden [272]
Het begrip "symbolen tijdens uitvoering" heeft betrekking op het totaal aantal geheugenvariabelen en database-velden dat in een dBASE-sessie is gebruikt. Het standaardaantal is 500. Tijdens een dBASE-sessie hebt u dit maximum overschreden en moet u dus minder symbolen gebruiken.

Een andere oplossing is in het bestand Config.db dit aantal te verhogen door de waarden die aan RTMAXBLKS en RTBLKSIZE zijn toegewezen, te wijzigen. Raadpleeg *Aan de slag met dBASE IV*.

Maximum aantal velden bereikt **[296]**
Het maximum aantal velden per database-bestand is 255.

Maximum aantal vensters is bereikt **[213]**
U hebt de beschikking over maximaal 20 vensters.

Maximum aantal verwijzingen naar symbolen in compileerfase overschreden **[273]***
Het begrip "symbolen in compileerfase" heeft betrekking op het totaal aantal unieke namen van geheugenvariabelen en database-velden in een gecompileerde procedure. Het standaardaantal compileerfasesymbolen is 500. Dit aantal is overschreden en moet dus worden verminderd.

Een andere oplossing is in het bestand Config.db dit aantal te verhogen door de waarde die aan CTMAXSYMS is toegewezen, te wijzigen. Raadpleeg *Aan de slag met dBASE IV*.

Maximumgrootte memoveld overschreden
De maximumgrootte van een memoveld bedraagt 64Kb.

Maximumveldbreedte overschreden
Dit is een foutmelding van het Control Center. Deze melding verschijnt wanneer u hebt geprobeerd een veld groter te maken dan de huidige maximumbreedte van het venster.

MDX-bestand is vol **[203]**
Het maximumaantal labels in een .mdx-bestand is 47.

MDX-bestand komt niet overeen met database **[207]**
Het indexbestand dat u wilde openen, komt niet overeen met de actieve database.

Meerdere sorteerinstructies in een kolom **[325]**
Tijdens een semantische controle is een kolom aangetroffen die meer dan één sorteerinstructie bevat (query-ontwerp).

Memo's zijn niet beschikbaar **[172]**
Deze foutmelding verschijnt wanneer u een .dbf-bestand met een bijbehorend .dbt-bestand opent en het .dbt-bestand niet wordt aangetroffen. Er wordt u gevraagd of u alle memogegevens wilt wissen. Wanneer u met N antwoordt, markeert dBASE IV deze gegevens zodat er geen memobewerkingen kunnen worden uitgevoerd. Als u daarna een memoveld probeert op te roepen, verschijnt deze melding.

Memovelden groter dan 64Kb worden niet door de tekstverwerker opgeslagen **[137]**
U hebt geprobeerd een memoveld op te slaan dat groter is dan de maximale 64Kb die voor de teksverwerker van dBASE IV geldt.

Memovelden kunnen geen prompt zijn **[280]**
U hebt geprobeerd een POPUP-venster te definiëren en daarbij een memoveld opgegeven als het veld in PROMPT FIELD<veldnaam>.

Mengen vergt te veel stappen **[278]**
Deze melding houdt verband met een fout tijdens het sorteren van een erg groot bestand. In de laatste fase van het sorteeralgoritme worden steeds meer grote blokken met gesorteerde

records gemengd. Deze melding wordt gegenereerd wanneer de zestiende stap is bereikt en de gesorteerde records nog steeds niet helemaal zijn gemengd. Er is een immens bestand nodig om deze grens te bereiken.

Menu is al in gebruik **[181]**
U hebt geprobeerd een MENU te activeren dat al is geactiveerd. Een menu kunt u niet tegelijkertijd tweemaal activeren.

MENU is niet geactiveerd **[178]**
Een menu moet u eerst activeren voordat u daaruit een optie kunt selecteren.

MENU is niet gedefinieerd **[168]**
Een menu moet u eerst definiëren voordat u dit kunt activeren.

Minimumveldbreedte bereikt
Dit is een foutmelding van het Control Center. Deze melding verschijnt wanneer u probeert een veld kleiner te maken dan de minimumbreedte voor dat veldtype.

MOVE of COPY niet mogelijk zonder selectie
U hebt geprobeerd gegevens te verplaatsen of te kopiëren zonder deze eerst te selecteren.

Naam moet een geldige veldnaam zijn
De veldnaam bevat te veel tekens of ongeldige tekens of spaties. Gebruik onderstrepingstekens in plaats van spaties. Een naam kan uit maximaal 10 tekens bestaan.

Naam van variabele kan niet langer zijn dan 10 **[312]**
De naam van de variabele die u in een QBE-bestandsstructuur hebt ingevoerd, is langer dan 10 tekens.

NDX-index mag niet aflopend zijn **[235]**
Gebruik voor het indexeren in afdalende volgorde een .mdx-index.

Neem eerst velden op in de visiestructuur **[321]**
U hebt voor een visie-query op F2 gedrukt maar er bevinden zich geen velden in de visiestructuur.

Netwerk-server niet beschikbaar **[148]**
De netwerk-server kan niet meer taken verwerken. Wacht totdat de netwerk-server uw taak kan uitvoeren. Raadpleeg de netwerkbeheerder.

Niet gevonden **[14]**
dBASE IV heeft de tekenreeks waarvoor u een zoekopdracht had gegeven, niet aangetroffen.

Niet gevonden **[82]**
Deze melding verschijnt bij Browse of Edit in een schermgrote bewerking wanneer een veld niet wordt aangetroffen.

Niet meer geheugen voor geheugenvariabelen beschikbaar **[21]**
Er is onvoldoende geheugenruimte in het systeem voor het definiëren van extra geheugenvariabelen.

Niet mogelijk! Er bestaat al een SQL-tabel met dezelfde naam **[374]**
U hebt geprobeerd een .dbf-bestand te maken maar er bestaat al een SQL-tabel met dezelfde naam. Kies een andere naam voor het database-bestand.

Niet onderbroken **[101]**
U hebt het commando RESUME gegeven terwijl de bestandsuitvoering niet was onderbroken.

Niet voldaan aan bewerkingsvoorwaarde **[171]**
Aan de WHEN-uitdrukking in het commando @...SAY, @...GET is niet voldaan.

Niet-beëindigd transactiebestand aanwezig, kan geen nieuwe transactie starten **[183]**
Er wordt een transactie uitgevoerd die nog niet is beëindigd. Verwijder met het commando END TRANSACTION de integriteitslabel uit de bestandsaanhef. Daarna kunt u een nieuwe transactie starten.

Niet-bestaande macrobibliotheek **[354]**
De macrobibliotheek die u hebt geprobeerd op te roepen, bevindt zich niet in de actieve bestandsindex of bestaat niet. Controleer de naam en de lokatie van de bibliotheek en probeer het opnieuw.

Niet-gedefinieerde fout
Deze melding verschijnt bij de DOS-foutmeldingen die niet zijn onderschept. dBASE IV onderschept nu ongeveer tien DOS-foutmeldingen. Als er een DOS-foutmelding moet worden weergegeven waarvoor dBASE IV geen equivalent heeft (een uitzonderlijke situatie), verschijnt bovenstaande melding op het scherm.

Niet-toegewezen bestandsnummer **[2]**
Deze melding kan verschijnen wanneer u hebt geprobeerd een I/O-bewerking uit te voeren met een bestandsverwerkingsnummer van een niet-geopend bestand. Dit is een interne fout die niet optreedt bij normaal gebruik van dBASE IV.

Numerieke overloop (met gegevensverlies) **[39]**
Als gevolg van een rekenkundige bewerking is de numerieke begrenzing van een veld of een geheugenvariabele overschreden.

ON PAD is al gedefinieerd voor dit keuzevenster **[170]**
Voor de aanwijzingsstrip die u in het laatste commando hebt benoemd, hebt u ON SELECTION PAD al eerder gedefinieerd. Voor één strip kunt u ON SELECTION PAD slechts eenmaal definiëren.

ON SELECTION is al gedefinieerd voor dit keuzevenster **[169]**
Voor de strip waarvoor u het commando ON SELECTION PAD opgeeft, is ON PAD al eerder opgegeven.

Onbekende functietoets **[104]**
De functietoets of de toetscombinatie die u wilde gebruiken, hebt u niet gedefinieerd.

Onbekende geheugenvariabele **[220]**
Er is een geheugenvariabele aanwezig die begint met een onderstrepingsteken maar die geen systeemgeheugenvariabele van dBASE IV is. Het onderstrepingsteken is gereserveerd voor systeemgeheugenvariabelen. Raadpleeg hoofdstuk 5 of de *Naslaggids dBASE IV* voor informatie over de systeemgeheugenvariabelen.

Onbekende parameter in commando **[36]***
Deze melding kan bij elk commando met sleutelwoorden of clausules worden weergegeven. Hiermee wordt aangeduid dat in een commandoregel een ongeldig sleutelwoord voorkomt.

Onbevoegde aanmelding **[132]**
U hebt geprobeerd zich aan te melden zonder geldig wachtwoord. Vraag de systeembeheerder u een geldig wachtwoord toe te wijzen.

Ongeldig aantal decimalen
Dit is een ontwerp-foutmelding van het Control Center. Het maximum aantal toegestane decimalen is 0 tot en met 18, dus in totaal 20 tekens inclusief het decimaalteken.

Ongeldig bestand met geheugenvariabelen **[55]**
U hebt geprobeerd een .mem-bestand terug te zetten, maar dBASE IV heeft in het gegevenstype van een van de variabelen een fout aangetroffen.

Ongeldig DIF-teken **[118]**
U hebt geprobeerd met APPEND gegevens toe te voegen uit een DIF-bestand dat een ongeldig teken of een besturingsteken bevat.

Ongeldig etiketbestand **[54]**
U hebt een etiketbestand (.lbl) uitgevoerd of gewijzigd dat niet met CREATE/MODIFY LABEL is gemaakt.

Ongeldig functie-argument **[11]**
Raadpleeg de beschrijving van de syntaxis voor functies in hoofdstuk 4, "Functies". U treft hierin de toegestane argumenten aan.

Ongeldig indexnummer **[106]**
De nummers van indexbestanden hebben een bereik tussen 0 en 7. U hebt geprobeerd voor SET ORDER TO een getal op te geven dat buiten dit bereik valt of waarvan de positie leeg is.

Ongeldig rapportbestand **[50]**
Het rapportbestand dat u hebt geprobeerd te activeren, heeft geen geldige indeling voor rapportbestanden. Controleer de bestandsnaam of maak met CREATE een nieuw rapportmodel.

Ongeldig schermbestand **[283]**
U hebt een indelingsbestand (.fmt) gewijzigd zonder het bijbehorende schermbestand (.scr) aan te passen. Het schermbestand komt niet overeen met het .fmt-bestand.

Ongeldig wachtwoord, ander wachtwoord a.u.b. **[153]**
U hebt het wachtwoord niet juist ingevoerd. Voer het wachtwoord opnieuw in. Als het probleem hierdoor niet wordt opgelost, raadpleegt u de systeembeheerder.

Ongeldige aanhef DIF-bestand **[115]**
Het DIF-bestand waaruit u hebt geprobeerd te importeren, bevat geen juiste bestandsaanhef.

Ongeldige aanhef SYLK-bestand **[119]**
Het bestand waaruit u hebt geprobeerd te importeren, bevat niet de juiste SYLK-aanhef.

Ongeldige aanhef van macrobestand **[356]**
Deze fout treedt op wanneer het ingelezen macrobestand beschadigd is of geen macrobestand van dBASE IV is. De melding betekent dat er tijdens het inlezen van de aanhef van het macrobestand een inconsistentie werd aangetroffen.

Ongeldige afmetingen van kader **[227]**
Wanneer u met het @-commando kaders tekent, moet rij 2 kolom 2 groter zijn dan rij 1 kolom 1.

Ongeldige begrenzing SYLK-bestand **[120]**
Het SYLK-bestand waaruit u hebt geïmporteerd, bevat gegevens die zich buiten de begrenzingen van het bestand bevinden.

Ongeldige bewerking van logisch veld **[90]**
U hebt geprobeerd het commando SORT of INDEX voor een logisch veld te gebruiken.

Ongeldige bewerking van memoveld **[34]**
U hebt geprobeerd het commando SORT of INDEX voor een memoveld te gebruiken.

Ongeldige bewerking. Database-structuur is leeg of gewijzigd **[362]**
Dit is een ontwerp-foutmelding van het Control Center. Deze melding verschijnt bij MODIFY STRUCTURE wanneer u hebt geprobeerd **F2** (EDIT/BROWSE) of **Shift-F2** uit te voeren nadat u de structuur van een database-bestand hebt gewijzigd.

Ongeldige datum **[81]**
U hebt geprobeerd een ongeldige datum in een datumveld in te voeren. Verwijder deze melding door op de spatiebalk te drukken.

Ongeldige DIF-indicator **[117]**
Het DIF-bestand waaruit u hebt geprobeerd te importeren, bevat een ongeldige aanduiding voor het gegevenstype.

Ongeldige DIF-vector - komt niet overeen met DBF-veld **[116]**
In het DIF-bestand waaruit u hebt geprobeerd te importeren, passen de bestandsaanhef en de gegevens niet bij elkaar.

Ongeldige extensie van bestandsnaam:<.ext> **[333]**
Voor het betreffende bestandstype hebt u een onjuiste extensie van de bestandsnaam opgegeven. Controleer welke extensie u voor dit bestandstype moet opgeven of geef geen extensie op, zodat de standaardextensie wordt gebruikt.

Ongeldige functienaam **[87]**
Raadpleeg hoofdstuk 4, "Functies", voor een bespreking van de functienamen.

Ongeldige groepnaam voor bestand, andere groepnaam a.u.b. **[159]**
De groepnaam die u in het PROTECT-programma hebt geprobeerd op te geven, is ongeldig. Geef een geldige groepnaam op.

Ongeldige indeling SYLK-bestand **[121]**
Het bestand waaruit u hebt geprobeerd te importeren, is geen bestand volgens de SYLK-indeling.

Ongeldige indexsleutel **[386]**
Met de indexsleutel kan geen index worden gemaakt. Zo is bijvoorbeeld het gebruik van door de gebruiker gedefinieerde functies (UDF) in een sleuteluitdrukking afhankelijk van de instelling van SET DBTRAP.

Ongeldige lengte numeriek veld
Een numeriek gegevensveld kan tussen 1 en 20 tekens lang zijn. U hebt geprobeerd deze lengte te overschrijden.

Ongeldige naam voor indexlabel:<labelnaam> **[284]**
Deze melding betekent dat de naam die u als labelnaam hebt opgegeven, niet in het .mdx-bestand voorkomt.

Ongeldige operator **[107]**
U hebt geprobeerd met een ongeldige operator gegevenstypen te koppelen.

Ongeldige optie bij DOS-commando SET **[99]**
Het bestand Autoexec.bat bevat een ongeldige DOS SET-optie. Raadpleeg de naslaghandleiding bij de MS-DOS of PC-DOS programmatuur.

Ongeldige printerpoort **[123]**
U hebt met het commando SET PRINTER TO een niet-bestaande poort opgegeven. Controleer welke poorten in de configuratie zijn opgenomen. Als u de uitvoer niet aan een andere poort hebt toegewezen, is de DOS-standaardpoort LPT1.

Ongeldige query in deze omgeving **[143]**
Dit is een foutmelding van het query-bestand. Deze melding verschijnt wanneer een veld waarnaar door het query-bestand wordt verwezen, zich niet in een van de actieve bestanden bevindt.

Ongeldige SET-uitdrukking **[231]**
Controleer de syntaxis van de ondersteunde SET-commando's.

Ongeldige structuur **[33]**
De structuur van het bestand voor het commando CREATE FROM is ongeldig.

Ongeldige toetslabel **[317]***
U hebt geprobeerd een functie toe te wijzen aan een toets die niet programmeerbaar is. Raadpleeg het schermgrote menu **SET** voor de lijst met de functietoetsen die u kunt programmeren.

Ongeldige veldbreedte
U hebt in het ontwerp van de database een ongeldige veldlengte voor het actieve veld ingevoerd.

Ongeldige versie van Lotus 1-2-3 spreadsheet **[297]**
U hebt geprobeerd een spreadsheet van Lotus te importeren dat niet uit versie 2 afkomstig is. Controleer de versie van het spreadsheet. Voordat u het spreadsheet probeert te importeren, converteert u het met het Lotus-hulpprogramma voor conversie naar versie 2.

Ongeldige waarde **[46]**
Bij het tekenen van een kader met het @-commando hebt u een TO-coördinaat opgegeven die kleiner is dan de coördinaat rij 1 kolom 1. Deze melding wordt ook bij SET TO-commando's weergegeven wanneer u een ongeldig aantal opgeeft.

Onjuist aantal parameters **[94]**
U hebt een .prg-programmabestand met een DO...WITH-constructie aangeroepen. Het sleutelwoord PARAMETERS in het programma, of het sleutelwoord WITH in het commando DO, wordt gevolgd door een onjuist aantal parameters.

Onjuist gebruik van macro **[266]***
U hebt bij de programmamacro (de &-functie) een geheugenvariabele gebruikt die niet van het type Teken is. Als u bijvoorbeeld aan een geheugenvariabele een numerieke waarde toewijst, zoals x = 3, geeft u x op bij de &-functie. X is eigenlijk een numerieke waarde, zodat deze niet kan worden gebruikt in een programmamacro.

Onjuiste array-afmetingen **[230]***
Deze melding betekent dat u een ongeldige waarde (zoals een nul) in de definitie van een reeks hebt opgegeven of dat u een reeks hebt gedefinieerd met meer dan 1023 elementen.

Onjuiste combinatie van gegevenstypen in uitdrukking
In het query-ontwerp hebt u een uitdrukking ingevoerd met conflicterende gegevenstypen.

Onjuiste omgeving voor het commando ROLLBACK **[193]**
Het database-bestand waarnaar wordt verwezen in de transaction-log, kan niet worden gevonden. Zoek het database-bestand of de database-bestanden die betrokken zijn bij de transactie en voer een ROLLBACK-commando uit voor elke .dbf-bestandsnaam. Nadat u met ROLLBACK de wijzigingen in alle .dbf-bestanden ongedaan hebt gemaakt, controleert u of de transaction-log is verwijderd. Als dat niet het geval is, gebruikt u QUIT of RUN om het logbestand uit het besturingssysteem te verwijderen.

Onjuiste PROCEDURE-naam **[32]***
U hebt geprobeerd een procedure te compileren, maar de procedurenaam wordt niet door dBASE IV herkend. Ga na of de procedurenaam uit geldige tekens bestaat of een geldige naam is.

Onjuiste tekenveldlengte
Dit is een ontwerp-foutmelding van het Control Center. U hebt in CREATE FROM een bestand dat met een structuur is uitgebreid, opgegeven en dit bestand heeft een veldlengte die zich niet binnen het bereik van 1 tot en met 254 bevindt.

Onvoldoende geheugen **[43]***
Uw opdracht kan niet worden uitgevoerd wegens onvoldoende beschikbaar RAM-geheugen. Vergroot de geheugenruimte of verwijder hulpprogramma's die permanent in het geheugen aanwezig zijn en start dBASE opnieuw.

Onvoldoende opslagruimte voor bewerking **[275]**
Er was onvoldoende ruimte op de schijf om de uitvoer van een procedure of van INDEX of SORT weg te schrijven. U vergroot de schijfruimte door bestanden van de vaste schijf te verwijderen of door een ander station op te geven. Als u een uitvoerbestand naar diskette wilde wegschrijven en het bestand overschreed de capaciteit van de diskette, kunt u de uitvoer niet naar diskette wegschrijven.

Onvoldoende records voor sorteerbewerking **[277]**
U kunt niet minder dan twee records sorteren.

Onvoldoende ruimte in rij
U hebt geprobeerd tekst in te voegen in een rij die vol was.

Onvoldoende ruimte in rij, veld afgekapt
U hebt geprobeerd een veld in een rij in te voegen die onvoldoende ruimte bood. Het veld is afgekapt zodat het in de beschikbare ruimte past.

Onvoldoende ruimte in rij, verplaats het veld met de cursortoetsen
U hebt geprobeerd een veld aan een regel toe te voegen die helemaal vol was. Er is onvoldoende ruimte om het veld na het eerste teken af te kappen.

Onvoldoende schijfruimte voor bewerking
De beschikbare schijfruimte moet voldoende zijn voor twee bestanden die elk dezelfde omvang hebben als het bronbestand. Geef als bestemming van het sorteerbestand een ander station op of vergroot de schijfruimte door bestanden te verwijderen.

Oorspronkelijk memo kan niet groter zijn dan 64Kb **[411]**
U hebt geprobeerd tekst toe te voegen aan een memo dat de maximale grootte van 64Kb al heeft bereikt.

Open eerst een database-bestand of visie **[281]**
U moet eerst met USE een bestand activeren voordat u daarop bewerkingen kunt uitvoeren.

Operator niet toegestaan bij rekenvelden **[313]**
U hebt een AVG-, COUNT-, GROUP BY-, MAX-, MIN- of SUM-operator in een QBE-rekenveldstructuur ingevoerd.

Operator niet toegestaan in doelbestand **[333]**
U hebt een AVG-, COUNT-, GROUP BY-, MAX-, MIN- of SUM-operator ingevoerd in een QBE-bestandsstructuur die het doelbestand voor een mutatie-query is.

Opgegeven printerpoort bestaat niet **[124]**
U hebt de af te drukken uitvoer aan een andere printerpoort toegewezen dan aan LPT1, maar de nieuwe poort bestaat niet.

ORDER TAG niet gevonden:<labelnaam> **[208]**
Deze melding duidt aan dat voor het commando SET ORDER TO een niet-bestaande label in een .mdx-bestand is opgegeven. Controleer de labelnamen van het actieve .mdx-bestand. Geef hierna het commando SET ORDER TO opnieuw.

Overschakelen naar Bladeren/Bewerken niet mogelijk zolang er fouten zijn **[344]**
Deze melding verschijnt wanneer u hebt geprobeerd Browse/Edit te selecteren en het query-ontwerp een fout bevat.

Pad of bestandsnaam ongeldig:<bestandsnaam>/<pad> **[202]**
De gebruikte bestandsspecificatie is onjuist.

PARAMETERS moeten aan begin van procedure worden opgegeven **[243]***
Van een gecompileerde procedure moet het commando PARAMETERS de eerste uit te voeren opdracht zijn. Bewerk het programmabestand en plaats het commando PARAMETERS aan het begin van het bestand.

PFS staat geen rijnummers boven de 20 toe
U kunt dit bestand niet naar PFS:FILE exporteren zonder het aantal rijen te verkleinen.

Pop-up is al in gebruik **[182]**
U hebt geprobeerd een pop-up te activeren, maar dit was al gebeurd. Een pop-up kan niet tweemaal zijn geactiveerd.

Pop-up is te klein **[287]**
Het pop-up venster is te smal voor selectiebalken. Dit is niet toegestaan. Definieer een groter venster en definieer de selectiebalken.

POPUP is niet geactiveerd **[179]**
U moet een pop-up menu eerst activeren voordat u een strip daarvan kunt selecteren.

POPUP is niet gedefinieerd **[165]**
U moet een pop-up menu eerst definiëren voordat u strips kunt definiëren of het menu kunt activeren.

Positie buiten venster **[288]**
De coördinaten die u (voor een strip of een aanwijzing of met een @-commando, ?-commando of een andere weergavecommando) hebt opgegeven, bevinden zich buiten de vensterbegrenzingen.

Positie van balk (BAR) moet een positief getal zijn **[167]**
De balken van een menu of een pop-up menu kunt u alleen met positieve getallen definiëren.

Printer is niet aangesloten of niet ingeschakeld **[126]**
U hebt geprobeerd uitvoer naar een printer te zenden die niet is aangesloten op de opgegeven poort, die niet online is of die is uitgeschakeld. Controleer de printer.

Printer is niet gereed **[125]**
U hebt geprobeerd uitvoer naar een printer te zenden die niet online is of die is uitgeschakeld. Controleer de printer.

Prioriteitsnummer voor sorteren kan niet hoger dan negen zijn
Dit is een foutmelding van het Control Center. Deze melding verschijnt wanneer u een prioriteitsnummer voor het sorteren invoert dat hoger is dan negen. Het is niet mogelijk op meer dan negen velden te sorteren.

Procedure is te omvangrijk (> 64Kb) **[258]***
Een gecompileerde procedure mag exact 65.520 bytes groot zijn. De betreffende procedure is te groot. Verdeel de procedure in kleinere functionele groepen.

Procedure niet gevonden:<procedurenaam> **[252]**
In de geopende programma- of procedurebestanden kan geen procedure met deze naam worden gevonden.

Procedures kunnen niet in een waarde resulteren **[385]***
Het resultaat van een procedure kan geen waarde zijn. Met een dBASE IV-functie of een door de gebruiker gedefinieerde functie kan het resultaat wel een waarde zijn.

Programmabestand was niet geladen **[91]**
Een binair programmabestand moet met LOAD worden geladen voordat u het met CALL in een programma kunt oproepen.

PROMPTS voor deze popup zijn al gedefinieerd **[279]**
U hebt met de optie PROMPT de inhoud van het pop-up menu gedefinieerd. Wanneer u de optie PROMPT hebt geselecteerd, kunt u de optie BAR niet meer toepassen.

Query is te complex **[338]**
Deze melding verschijnt als de totale breedte van de velden in een unieke query meer dan 100 tekens bedraagt of als de commandoregel die door het query-mechanisme wordt gegenereerd, meer dan 1024 tekens bevat.

Als het totale breedte van alle geselecteerde velden in de visiestructuur meer dan 100 tekens bedraagt, moet u een of meer velden uit de visiestructuur verwijderen.

Rechtermarge moet kleiner dan of gelijk aan 255 zijn **[225]**
De waarde van de instelling voor de rechtermarge kan niet groter zijn dan de waarde van de paginabreedte.

Record buiten bereik **[5]**
U hebt geprobeerd de cursor naar een niet-bestaand record te verplaatsen.

Record in gebruik door<gebruikersnaam> **[373]**
Het record dat u wilde activeren, is door een andere gebruiker in het netwerk, die in de foutmelding wordt genoemd, vergrendeld.

Record in gebruik door andere gebruiker **[109]**
Het bestand dat u wilde activeren, wordt momenteel door een andere gebruiker in het netwerk gebruikt.

Record niet in index **[20]**
U hebt geprobeerd de cursor naar een record te verplaatsen dat niet in de index voorkomt. Maak voor dit bestand een nieuwe index of gebruik geen indexbestand.

Record niet ingevoegd **[25]**
Het is niet gelukt het record met het commando INSERT tussen te voegen. Probeer het record opnieuw tussen te voegen.

Regel bevat meer dan de maximale 1024 tekens **[18]***
Een commandoregel mag uit niet meer dan 1024 tekens bestaan.

Regelnummer moet tussen 0 en paginalengte zijn **[222]**
Controleer het bereik van de toewijzingen van de regelnummers en controleer of het regelnummer binnen de paginalengte van de geheugenvariabele past.

Rekenveld moet een uitdrukking bevatten
U hebt geprobeerd een rekenveld zonder uitdrukking te maken. Het submenu voor validatie was actief en u hebt op **Ctrl-End** gedrukt maar geen uitdrukking ingevoerd.

Rekenvelden in een visie behoren niet leeg te zijn **[323]**
U hebt de uitdrukking van een rekenveld in de visie verwijderd.

Relatierecord is in gebruik door een andere gebruiker **[142]**
Het record in het gerelateerde database-bestand is vergrendeld door een andere gebruiker.

REPLICATE(): tekenreeks te lang **[88]**
De tekenreeks die u hebt geprobeerd te kopiëren, bestaat uit meer dan 255 tekens.

RETURN TO is ongeldig in een door de gebruiker gedefinieerde functie **[383]***
Het commando RETURN resulteert in een waarde en beëindigt alle door de gebruiker gedefinieerde functies. Vanuit een door de gebruiker gedefinieerde functie kan met RETURN TO niet worden teruggekeerd naar een ander programma of een andere procedure.

ROLLBACK is niet mogelijk binnen een transactie **[197]**
Het commando ROLLBACK in combinatie met een naam van een database-bestand kan alleen na een apparatuurstoring worden gebruikt. Als u tijdens de uitvoering van een transactie het commando ROLLBACK met een bestandsnaam opgeeft, verschijnt deze melding.

RunTime-opdracht ongeldig: weggelaten uit .DBO **[348]**
Er is een commando aangetroffen dat niet kan worden gebruikt in RunTime. Dit commando wordt niet gecompileerd in het .dbo-bestand.

Schijf geeft I/O-fout
De fout die is opgetreden bij het lezen van of het wegschrijven naar de schijf kan een van de volgende oorzaken hebben: een volle schijf, problemen met de apparatuur of media die niet bij elkaar passen.

Schijf vol tijdens opslaan van bestand:<bestandsnaam> **[56]**
of
Schijf %1s: is vol. Verwijder oude bestanden van schijf.
De vaste schijf of de diskette waarnaar u hebt geprobeerd weg te schrijven, is vol. Vergroot de beschikbare ruimte op de vaste schijf door bestanden te verwijderen of gebruik een nieuwe diskette die op de juiste manier is geformatteerd.

Schrijffout in station<station>
Dit is een foutmelding van het besturingssysteem die aanduidt dat er een schrijffout is opgetreden in het station dat in de melding wordt vermeld.

Slechts één mutatie per query toegestaan **[328]**
Bij het verplaatsen van de cursor uit het veld onder de bestandsnaam van een bestandsstructuur is gebleken dat er meer dan één mutatie-operator (voor vervangen, toevoegen, wissen of wismarkering verwijderen) is opgegeven (query-ontwerp).

Sorteerinstructie komt meer dan eenmaal voor **[322]**
Tijdens een semantische controle heeft dBASE IV identieke sorteerinstructies in twee verschillende kolommen aangetroffen (query-ontwerp).

Sorteerinstructie moet overeenkomen met index **[330]**
U hebt een sorteeroperator ingevoerd in een kolom van een samengestelde index in een QBE-bestandsstructuur, maar de sorteerrichting komt niet overeen met de indexrichting. Als de index bijvoorbeeld afdalend is, is het niet mogelijk om ASC1 in te voeren.

Sorteervolgorde moet O of A zijn
Bij het selecteren van de sorteervolgorde in het database-ontwerp hebt u voor Database sorteren op veldenlijst op de verkeerde toets gedrukt.

SPACE(): negatief **[60]**
Voor de functie SPACE() is een negatief getal ingevoerd.

SPACE(): te groot **[59]**
Voor de functie SPACE() is een getal ingevoerd dat groter is dan 254.

Sprongopdracht moet eindigen voor eerste commando @ **[308]***
In een indelingsbestand kan het eerste initialisatiegedeelte een willekeurig commando bevatten, maar een besturingsstructuur zoals IF moet met ENDIF worden afgesloten voordat het tweede gedeelte, dat alleen @-commando's bevat, begint. Controleer het indelingsbestand en sluit de sprongopdrachten van het programma af voordat het gedeelte met de @-commando's begint.

SQL-fout tijdens uitvoering **[335]**
Deze melding verschijnt wanneer tijdens de uitvoering van een SQL-opdracht een fout optreedt.

SQL-tabel kan niet worden gewijzigd **[375]**
U kunt een SQL-tabel niet in de dBASE IV-modus, maar wel in de interactieve SQL-modus wijzigen.

SQRT(): negatief **[61]**
De invoer voor de vierkantswortelfunctie mag geen negatief getal zijn.

Station of catalogus bevat geen bestanden van het gevraagde type **[53]**
U hebt in de bestandsindex met jokers gezocht naar een bestandstype uit DOS maar er zijn geen bestanden die aan de filtervoorwaarden voldeden.

STORE: tekenreeks te lang **[79]**
Tekenreeksen mogen uit niet meer dan 254 tekens bestaan.

STR(): buiten bereik **[63]**
Deze foutmelding verschijnt wanneer de waarde van het derde argument in de functie STR() de waarde van het tweede argument overschrijdt.

Strip (PAD) is niet gedefinieerd **[164]**
U moet een strip definiëren voor pop-up menu's en andere menu's.

Strip (PAD) is niet gedefinieerd voor dit MENU **[180]**
U hebt geprobeerd een menu te activeren dat wel is gedefinieerd maar dat geen strips bevat. Voordat u een menu kunt activeren, moet u een strip definiëren. Een menu met alleen een naam kan niet worden weergegeven.

STUFF(): tekenreeks te lang **[102]**
De tekenreeks die u in een veld hebt geprobeerd tussen te voegen, past niet in dat veld.

SUBSTR(): beginpunt buiten bereik **[62]**
Deze melding verschijnt als het tweede argument kleiner is dan 0 of langer is dan de invoer-tekenreeks.

Syntaxisfout **[10]***
Het query-ontwerp bevat een syntaxisfout.

Syntaxisfout in definitie van query
Deze foutmelding verschijnt wanneer u een query laadt die een syntaxisfout bevat, bijvoorbeeld scheidingstekens die niet met elkaar overeenkomen of een ontbrekende komma.

Syntaxisfout in inhoudsuitdrukking
Deze foutmelding wordt in de etikettengenerator weergegeven. De melding geeft aan dat het etiketmodel van dBASE III PLUS een ongeldige uitdrukking bevat.

Systeemfout
Deze melding wordt door het besturingssysteem weergegeven ter aanduiding van fouten in de apparatuur bij het lezen van of schrijven naar schijf. Controleer de compatibiliteit van de media: de schijf is mogelijk op een niet-compatibele manier geformatteerd. Controleer de bestandsspecificatie voor het doelbestand zodat u er zeker van bent dat de naam van het station waarnaar u wilt wegschrijven, daarin is opgenomen.

Tabel is vol **[105]**
U kunt maximaal 16 binaire programmabestanden in een CALL-tabel laden die u vervolgens vanuit dBASE IV kunt aanroepen. Verklein het aantal bestanden.

Tabstops moeten in oplopende volgorde worden opgegeven **[226]**
De waarden die aan de geheugenvariabele *_tabs* zijn toegekend, moeten in oplopende volgorde staan.

TAG niet gevonden **[209]**
Deze foutmelding verschijnt als u een label opgeeft die niet in het geopende .mdx-bestand wordt aangetroffen.

Te veel @A-30-commando's op één pagina van indelingsbestand
Deze melding wordt bij het exporteren naar PFS:FILE gegenereerd en duidt aan dat er te veel @-commando's zijn. Hierdoor kan er niet naar de PFS:FILE-indeling worden geëxporteerd. Op één pagina bevinden zich meer dan 200 @-commando's. Verklein het aantal @-commando's.

Te veel bestanden geopend **[6]**
U hebt geprobeerd het aantal bestanden dat u in het bestand Config.sys met het commando FILES= hebt ingesteld, te overschrijden. Sluit enkele bestanden of sluit alle bestanden en verlaat dBASE IV zodat u in het bestand Config.sys de waarde op de regel FILES= kunt vergroten. Wanneer u het bestand Config.sys hebt gewijzigd, start u de computer opnieuw op, zodat met de nieuwe waarde rekening wordt gehouden.

Te veel indexen **[28]**
Naast het bijbehorend .mdx-bestand kunt u maximaal 10 .ndx-bestanden en .mdx-bestanden in een werkgebied openen.

Te veel pagina's in indelingsbestand
Deze melding wordt bij het exporteren naar PFS:FILE weergegeven wanneer er zich te veel pagina's in het indelingsbestand bevinden die naar de PFS:FILE-indeling moeten worden gexporteerd.

Een indelingsbestand mag uit maximaal 32 pagina's bestaan. Wanneer u meer pagina's nodig hebt, verdeelt u het bestand in twee of meer indelingsbestanden.

Te veel velden die als sorteersleutel dienen **[276]**
U mag tien sorteersleutelvelden kiezen.

Te veel velden in WK1-bestand **[391]**
Het Lotus-werkblad dat u hebt geprobeerd te importeren, bevat meer velden dan is toegestaan.

Te veel WITH-operatoren **[319]**
Tijdens een semantische controle zijn te veel WITH-operatoren die voor de REPLACE-operator zijn opgegeven, in de bestandsstructuur aangetroffen (query-ontwerp).

Tekenreeks niet beëindigd **[35]***
Een tekenreeks moet worden afgesloten met een geldig dBASE IV-scheidingsteken. De scheidingstekens zijn: enkele aanhalingstekens, dubbele aanhalingstekens en rechte haken.

Tekenreeks niet gevonden
Met de zoekbewerking is de opgegeven tekenreeks niet aangetroffen.

Tijdens een transactie kan de database niet worden gesloten **[185]**
Voordat u een database tijdens een transactie kunt sluiten, moet u de transactie helemaal uitvoeren en het commando END TRANSACTION geven, of ROLLBACK uitvoeren.

Tijdens een transactie kan dit commando niet worden uitgevoerd **[186]**
Bij de beschrijving van het commando BEGIN TRANSACTION treft u een lijst aan met de commando's die u tijdens een transactie niet kunt geven.

Tijdens een transactie kunnen indexbestanden niet worden gesloten **[187]**
Voordat u indexbestanden tijdens een transactie kunt sluiten, moet u de transactie helemaal uitvoeren en het commando END TRANSACTION geven, of ROLLBACK uitvoeren.

Toetsaanslag zou leiden tot lus in macro:<macronaam> **[363]**
Wanneer u toetsaanslagen voor een bepaalde toets vastlegt en u op die toets drukt, verschijnt deze foutmelding en wordt de toetsaanslag niet vastgelegd. Als deze wel zou worden opgeslagen, zou dit een lusbewerking kunnen veroorzaken. Achter de dubbele punt staat de naam van de macro.

Toevoegen in kolomvolgorde niet mogelijk **[147]**
U hebt geprobeerd met APPEND een spreadsheet van MultiPlan (in de SYLK-indeling) toe te voegen, maar de rijen zijn niet in oplopende volgorde gerangschikt. Controleer of de kolommen van het spreadsheet op de juiste manier zijn geordend voordat u dit commando opnieuw geeft.

Totale grootte van velden in visie te groot voor unieke query **[381]**
Deze melding verschijnt als de totale omvang van de velden in een unieke query groter is dan 100 tekens. Aangezien de totale breedte van de geselecteerde velden niet meer dan 100 tekens mag bedragen, selecteert u alleen de velden die nodig zijn om de query uniek te maken.

Transacties kunnen niet worden genest **[198]**
U kunt transacties niet binnen andere transacties nesten. Geef het commando END TRANSACTION voordat u een nieuwe transactie start.

TRAP kan niet worden uitgeschakeld terwijl Debugger actief is **[390]**
Voordat u het debugger-programma aanroept, moet u SET TRAP op ON of OFF instellen.

U kunt alleen een relatie leggen tussen velden
U hebt de optie **Koppeling leggen door aanwijzen** geselecteerd (in het query-ontwerp) maar de cursor bevond zich in het veld onder de bestandsnaam van een bestandsstructuur.

Uitdrukking ontbreekt **[152]***
In de syntaxis van een commando zoals DO WHILE of van een IF-opdracht ontbreekt een uitdrukking.

Valutateken kan geen cijfers bevatten **[295]**
Gebruik voor de definitie van een valutateken alleen tekens.

Variabele niet gevonden:<naam variabele> **[12]**
De opgegeven variabele is geen veldnaam of is niet in het actieve, geselecteerde werkgebied of in het met ALIAS opgegeven werkgebied aanwezig. Ook bestaat het memobestand dat u hebt geprobeerd te kopiëren of waaraan of waaruit u gegevens wilt toevoegen, niet.

In het query-ontwerp is door dBASE IV een niet-gedefinieerde variabele aangetroffen.

Veld moet een naam hebben
U hebt geprobeerd een rekenveld of een samenvattingsveld dat geen naam heeft, te verbergen. Verborgen velden moeten een naam hebben.

Veld moet memoveld zijn **[350]**
U hebt bij het commando APPEND/COPY MEMO<veldnaam> een veldnaam ingevoerd die niet de naam van een memoveld is.

Veld niet gevonden:<veldnaam> **[48]**
U hebt een zoekopdracht voor een veld gegeven maar dit veld is in het gebruikte bestand niet aangetroffen.

Veldenlijst te gecompliceerd **[141]***
Verminder het aantal velden in de veldenlijst.

Veldnaam is noodzakelijk
U hebt geen veldnaam opgegeven bij het uitvoeren van het commando CREATE of MODIFY STRUCTURE.

Veldnaam komt meer dan eenmaal voor
U hebt bij het maken of wijzigen van een database-structuur een veldnaam ingevoerd die al in de structuur voorkomt.

Veldnaam niet gevonden
In de sorteerlijst voor velden hebt u een veldnaam ingevoerd die niet in het actieve database-bestand voorkomt.

Veldnaam te lang
De nieuwe naam van het veld in de visiestructuur bestaat uit te veel tekens. De veldnaam wordt achter het tiende teken afgekapt.

Veldtype moet T, N, Z, D, L of M zijn
U hebt op de verkeerde toets gedrukt terwijl u het gegevenstype wijzigde.

Venster is niet gedefinieerd **[214]**
U hebt geprobeerd een venster te activeren dat niet is gedefinieerd. Definieer eerst het venster.

Venster is te klein **[285]**
U hebt een venster gedefinieerd dat te klein is voor BROWSE. Het venster moet vier rijen hoog zijn en breed genoeg zijn voor ten minste één veld.

Venster is vergrendeld:<vensternaam> **[184]**
Bij een onderbrekend commando, zoals in een UDF- of ON-routine, is RELEASE WINDOW of CLEAR WINDOW niet mogelijk voor een venster dat tijdens de onderbreking is geopend.

Venster voor commandobewerking; met CTRL-END gaat u naar de stip
Dit is een melding van de tekstverwerker voor de commando's in het zoom-venster.

Venster wordt nu niet weergegeven **[215]**
Het venster dat u in DEACTIVATE WINDOW<vensternaam> hebt opgegeven, is niet actief op het scherm aanwezig.

Vergrendelingstabel is vol **[217]**
U hebt alle beschikbare recordvergrendelingen gebruikt. Het maximum aantal is 50.

Verwijderen groep is alleen van invloed op groepzones
Deze melding verschijnt wanneer u een zone die geen deel van een groep uitmaakt, markeert en daarna uit het menu Zones de optie **Verwijderen groep** selecteert.

Visie kan niet meer velden bevatten (255 is maximum) **[353]**
U hebt geprobeerd aan de visie een 256ste veld toe te voegen.

Volgorde moet worden opgegeven met indexlabel of bestandsnaam **[329]**
U hebt voor het commando SET ORDER TO een getal ingevoerd maar er is alleen een actief .mdx-bestand. U moet een naam van de indexlabel opgeven. U kunt alleen een getal invoeren bij .ndx-bestanden.

Voor dit pop-up menu zijn geen balken gedefinieerd **[166]**
Voor elk pop-up menu moet u balken definiëren.

Voor kopiëren van bijbehorend .MDX is één vrij werkgebied noodzakelijk **[487]**
Als u de optie PRODUCTION van het commando COPY gebruikt, moet er één werkgebied beschikbaar zijn. Sluit het database-bestand in het andere werkgebied en voer het commando COPY opnieuw uit.

Voorwaardenkader kan niet verder worden uitgebreid
Het onderste gedeelte van het voorwaardenkader is bereikt.

Waarde van tabstop overschrijdt 255 **[399]**
De waarde van een tabstop van de tekstverwerker kan niet groter zijn dan 255.

Wachtwoord en bevestiging komen niet overeen **[154]**
Het eerste wachtwoord dat u hebt getypt, verschilt van het tweede wachtwoord. Typ het wachtwoord opnieuw.

Wachtwoordbestand is door een ander in gebruik **[349]**
Het wachtwoordbestand wordt door een ander gebruikt. Probeer het later opnieuw of raadpleeg de systeembeheerder.

Weergavebreedte groter dan 255 **[400]**
De weergavebreedte voor rapporten kan niet groter dan 255 tekens zijn.

Wegschrijven naar database onmogelijk vanwege onvoltooide transactie **[201]**
Door een onvolledige transactie is de database gemarkeerd ter aanduiding dat deze in gebruik is. Als u geen transactie hebt gestart, controleert u met de functie COMPLETED() of de database bij een transactie is betrokken. Als het resultaat van deze functie onwaar (.F.) is, wacht u totdat de transactie helemaal is beëindigd. Daarna geeft u het commando END TRANSACTION zodat de wijzigingen worden opgeslagen, of het commando ROLLBACK zodat de wijzigingen worden geannuleerd en de integriteitsvlag uit het database-bestand wordt gewist.

Wegschrijven naar transaction-log niet mogelijk **[188]**
Het gewenste log-bestand van de transactie is niet geopend of beschikbaar. Annuleer met het commando END TRANSACTION de transactie die u hebt geprobeerd uit te voeren.

Werkgebied gereserveerd door SQL [219]
U kunt een bestand niet in een werkgebied activeren als dit werkgebied al voor SQL is gereserveerd.

Werkgebied is al gerelateerd [298]
Een serie relaties kan slechts eenmaal naar een database-bestand in een werkgebied verwijzen. In een serie relaties kunt u een database-bestand niet tweemaal definiëren. Als u dit wel doet, wordt deze melding weergegeven.

Werkgebied is vergrendeld:<aliasnaam> [409]
Er wordt een commando uitgevoerd in het actieve werkgebied en met een onderbrekend commando (wellicht in een UDF- of ON-routine) wordt een poging gedaan het database-bestand in het werkgebied te sluiten. Het database-bestand moet geopend blijven, zodat de bewerking kan worden voortgezet met het eerste commando.

Wijzigen groep is alleen van invloed op groepzones
Deze foutmelding verschijnt wanneer een gebruiker een zone die geen onderdeel van een groep uitmaakt, markeert en daarna uit het menu **Zones** de optie **Wijzigen groep** selecteert.

<veld> gegevenstype is onjuist<veld>
Deze melding is het resultaat van een controle op de consistentie van de velden in het formulier, rapport of etiket en het actieve database-bestand.

<veld> niet gevonden in<database-bestand>
Deze melding is het resultaat van een controle op consistentie van de velden in het formulier, rapport of etiket en het actieve database-bestand.

-: aaneengeschakelde tekenreeks te lang [76]
+: aaneengeschakelde tekenreeks te lang [77]
De tekenreeks die het resultaat is van de aaneenschakeling van twee tekenreeksen, bestaat uit meer dan 254 tekens. Het verschil tussen de twee meldingen is de aaneenschakelingsoperator waarmee de grote tekenreeks is gemaakt: een + of een -.

**** Let op ** Indexen konden niet worden bijgewerkt** [378]
dBASE IV kon de bij de actieve database behorende indexbestanden niet bijwerken omdat deze indexbestanden niet werden aangetroffen of omdat deze indexen niet met USE waren geactiveerd.

**** LET OP ** Onvoltooide transactie gevonden** [190]
U hebt geprobeerd het commando BEGIN TRANSACTION te geven zonder eerst het commando END TRANSACTION op te geven.

**** LET OP ** Waarschijnlijk zullen gegevens verloren gaan. Akkoord (J/N)** [70]
Deze melding verschijnt nadat u bent gewaarschuwd dat een schijf vol is. U kunt andere bestanden van schijf verwijderen of de bewerking annuleren.

^ of **: negatief grondtal, breuk in exponent [78]
U mag bij rekenkundige en trigonometrische functies geen negatief getal als parameter invoeren. Exponenten moeten gehele getallen zijn.

^—- Afgekapt [74]
Een regel met tekst in het bestand Config.db was te lang of niet beëindigd en is daarom afgekapt. Een tekenreeks in Config.db mag uit maximaal 255 tekens bestaan.

^—- Buiten bereik [75]
De parameter bij deze functie overschrijdt het bereik. Raadpleeg hoofdstuk 4 “Functies” voor het bereik van de functie.

^— Sleutelwoord niet gevonden **[86]**
U hebt in het bestand Config.db een sleutelwoord opgegeven dat in dBASE IV ongeldig is. Raadpleeg Aan de slag met dBASE IV voor een lijst met geldige sleutelwoorden.

^^ Alleen ON of OFF mogelijk **[73]**
Het SET-commando dat u hebt gewijzigd, kan alleen ON of OFF als instelling van de parameter hebben.

20 rekenvelden is het maximum
U hebt geprobeerd aan de structuur van een rekenveld in QBE een eenentwintigste rekenveld toe te voegen.

Foutmeldingen van SQL

In deze bijlage treft u een alfabetische lijst aan van de foutmeldingen die bij de SQL-commando's van dBASE IV kunnen worden weergegeven. De nummers van de foutmeldingen kunnen met de functie ERROR() worden onderschept en staan tussen rechte haken rechts van de foutmelding.

OPMERKING
Als er tijdens de uitvoering van een SQL-programma fouten optreden, volgt u het advies op dat bij de desbetreffende foutmelding wordt vermeld. Als de fout daarna opnieuw optreedt, kunt u als volgt te werk gaan:

1. *Controleer met DBCHECK of het database-bestand en de indexstructuren overeenkomen met de ingevoerde gegevens in de bijbehorende SQL-catalogustabellen.*
2. *Werk met RUNSTATS de statistische gegevens in de SQL-catalogustabellen bij.*
3. *Compileer de programma's waarin de fout is opgetreden, opnieuw.*

*** alleen toegestaan voor functie COUNT [1082]**
U hebt voor een andere verzamelfunctie dan COUNT een asterisk als argument gebruikt. Het argument voor de functies AVG(), MAX(), MIN() en SUM() moet een kolomnaam zijn.

Aanhalingstekens aan einde van tekenreeks ontbreken [1003]
Tekenreeksen moeten tussen aanhalingstekens worden geplaatst. Voeg het ontbrekende aanhalingsteken (") toe. Het is ook mogelijk dat u op een verkeerde plaats een aanhalingsteken hebt ingevoegd, dat is geïnterpreteerd als de beginmarkering van een tekenreeks. Controleer ook of u enkele (') en dubbele (") aanhalingstekens door elkaar hebt gebruikt.

Aantal decimalen komt niet overeen met tabel/kolom:<tabelnaam en kolomnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat de decimalen die zijn aangegeven in de kolom Colscale van de catalogustabel Syscols, niet overeenkomen met de decimalen van de werkelijke kolom. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-bestand en het .mdx-bestand vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Aantal indexen komt niet overeen met tabel:<tabelnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat het aantal indexen dat wordt aangegeven in de catalogustabel Systabls niet overeenkomt met het werkelijke aantal indexen. Kopieer eerst het .dbf-bestand en de desbetreffende .mdx-bestanden naar een andere bestandsindex en verwijder de tabel (met DROP) uit de huidige bestandsindex. Kopieer het .dbf-bestand en de .mdx-bestanden vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE uit om de bestanden opnieuw te definiëren. Voer daarna het commando DBCHECK of RUNSTATS opnieuw uit.

Aantal kolommen in indexsleutel komt niet overeen met tabel/index:<tabelnaam en indexnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat het aantal kolommen in de indexsleutel, zoals dit is vastgelegd in de catalogustabel Sysidxs, afwijkt van het werkelijke aantal kolommen in de index. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-bestand en het .mdx-bestand vervolgens

terug naar de oorspronkelijke database-bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Aantal kolommen in visie komt niet overeen met aantal geselecteerde kolommen [1104]
Het aantal kolommen dat u hebt opgegeven in de kolommenlijst van het commando CREATE VIEW, komt niet overeen met het aantal kolommen in het resultaat van de clausule SELECT. Als alle kolommen in de visie benoemde kolommen zijn (en niet het resultaat van verzamelfuncties, uitdrukkingen, enzovoort), is een kolommenlijst niet vereist. In dat geval krijgen de kolommen in de visie dezelfde namen als in de clausule SELECT.

Aantal kolommen komt niet overeen in tabel:<tabelnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat het aantal kolommen in een tabel niet overeenkomt met het aantal dat is aangegeven in de kolom Colcount van de SQL-catalogustabel Systabls. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-bestand en het .mdx-bestand vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Aantal kolommen moet gelijk zijn bij UNION-bewerking [1092]
De SELECT-opdrachten die zijn samengevoegd met het sleutelwoord UNION, resulteren niet in hetzelfde aantal kolommen. Wijzig de clausules van de SELECT-opdrachten zodanig dat de opdrachten in hetzelfde aantal kolommen resulteren. De kolommen moeten ook van hetzelfde gegevenstype zijn en dezelfde lengte hebben.

Aantal kolommen na SAVE TO TEMP niet gelijk aan aantal SELECT-kolommen [1148]
De kolommenlijst die u na SAVE TO TEMP opgeeft, moet hetzelfde aantal kolommen bevatten als de resultaten van de SELECT-opdracht die worden opgeslagen. Als alle kolommen in de resultatentabel benoemde kolommen zijn (die niet zijn afgeleid van functies, constanten, enzovoort) hoeft u in de clausule SAVE TO TEMP geen kolommenlijst op te geven.

Aantal variabelen in clausule INTO niet gelijk aan aantal SELECT-onderdelen [1119]
Een SELECT-opdracht in een .prs-programma bevat een INTO-clausule waarvan het aantal variabelen niet overeenkomt met het aantal kolommen in de SELECT-clausule.

Alias-naam bestaat al:<aliasnaam> [1083]
U hebt de aliasnaam al voor een andere tabel of visie in dezelfde clausule gebruikt. Geef een andere aliasnaam op.

Alle SELECT-kolommen moeten in een verzamelfunctie worden opgenomen [1164]
Wanneer u HAVING zonder GROUP BY gebruikt, moeten alle SELECT-kolommen constanten zijn of in een SQL-verzamelfunctie zijn verwerkt. Neem alle kolomnamen in een verzamelfunctie op, verwijder deze uit de SELECT-clausule of voeg een GROUP BY-clausule toe.

ALTER niet toegestaan met visies:<visienaam> [1151]
In een ALTER TABLE-opdracht kunt u alleen basistabellen en bijbehorende synoniemen gebruiken. Een visie kunt u met DROP wissen en vervolgens opnieuw definiëren zodat daarin extra kolommen die zijn toegevoegd aan de basistabel, kunnen worden opgenomen.

ALTER-recht niet ontnomen
De gebruiker die dit recht ontneemt, heeft dit nooit toegekend aan de gebruiker aan wie het wordt ontnomen, of het recht is al ontnomen.

ALTER-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht ALTER of heeft dit niet met WITH GRANT OPTION verworven.

Argument te lang in CREATE INDEX/GROUP BY/ORDER BY/UNION/SELECT DISTINCT **[1185]**
Door een indexuitdrukking van meer dan 100 tekens is een fout opgetreden in het commando CREATE INDEX of in een SELECT-commando met clausules (zoals GROUP BY of ORDER BY) die gebruik maken van indexen. Geef voor het commando CREATE INDEX een kortere indexsleutel op. Als deze fout in een SELECT-commando is opgetreden, betreft het een interne fout. U kunt de clausule GROUP BY of ORDER BY verwijderen of het aantal kolommen in DISTINCT of UNION verminderen.

Bestand heeft geen SQL-codering **[1227]**
Door DBCHECK of RUNSTATS is een gecodeerd bestand aangetroffen dat niet met de SQL-codeersleutel is gecodeerd. Het bestand moet door de gebruiker worden gedecodeerd of worden verwijderd voordat het commando opnieuw wordt uitgevoerd.

Bestand heeft ongeldige SQL-codering **[1228]**
Door DBCHECK of RUNSTATS is een bestand met een ongeldige SQL-codering aangetroffen. Probeer met het commando UNLOAD van SQL een ongecodeerde versie van het bestand te maken. Als dat lukt, wist u de gecodeerde versie, geeft u het commando DBDEFINE<bestandsnaam> en voert u het commando DBCHECK of RUNSTATS opnieuw uit. Als UNLOAD geen succes heeft, moet u de tabel in SQL met DROP wissen voordat u DBCHECK of RUNSTATS kunt uitvoeren.

Bestand is gecodeerd **[1226]**
DBDEFINE kan niet op een gecodeerd bestand worden uitgevoerd. Als het bestand een dBASE-codering heeft, stelt u SET SQL op OFF in. Met het dBASE-commando COPY TO maakt u een ongecodeerde versie van het bestand. Wis de gecodeerde versie voordat u DBDEFINE opnieuw uitvoert. Als het bestand een SQL-codering heeft, maakt u met het SQL-commando UNLOAD een ongecodeerde versie van het bestand. Wis de gecodeerde versie voordat u DBDEFINE opnieuw uitvoert.

Bestand niet gevonden in actieve database **[1231]**
Tijdens de uitvoering van het commando DBCHECK, DBDEFINE of RUNSTATS is een fout opgetreden omdat het bestand niet in de actieve bestandsindex van de database werd aangetroffen. Controleer of de bestandsnaam geen typefouten bevat en of het benoemde .dbf-bestand zich in de actieve bestandsindex van de database bevindt.

Bestandsnaam is identiek aan bestaand synoniem **[1227]**
Deze foutmelding wordt bij DBDEFINE weergegeven wanneer de naam van het .dbf-bestand dat in het commando is opgegeven, gelijk is aan een bestaand SQL-synoniem. Geef het .dbf-bestand een nieuwe naam voordat u DBDEFINE<bestandsnaam> opnieuw uitvoert, of verwijder het synomiem.

Bestand SYSDBS in bestandsindex van SQL kan niet worden geopend **[1249]**
Het bestand SYSDBS is niet aanwezig of beschadigd.

Bij deze bewerking kan het bestand niet in dBASE-modus worden geopend **[2011]**
Bij het gebruiken van een commando voor het definiëren van gegevens, zoals CREATE of DROP, zijn de bijbehorende tabel of systeemtabellen in de dBASE-modus geopend. Sluit de tabellen met CLOSE of USE voordat u verder gaat.

Bij UNION moet(en) ORDER BY-kolom(men) worden aangegeven met gehele getallen [1096]

Met een ORDER BY-clausule kunt u de resultatentabel van twee of meer SELECT-opdrachten die met UNION zijn samengevoegd, ordenen. U moet echter gehele getallen gebruiken in plaats van kolomnamen, aangezien de kolomnamen in alle SELECT-clausules van elkaar kunnen verschillen. Vervang de kolomnamen door gehele getallen (om bijvoorbeeld op de eerste kolom te ordenen, gebruikt u 1, om op de tweede kolom te ordenen, gebruikt u 2, enzovoort).

Catalogustabel Sysdbs bestaat niet [1229]

De tabel Sysdbs moet zich in de SQLHOME-bestandsindex bevinden. Controleer of u het bestand Sysdbs.dbf niet hebt gewist en of de SQLHOME-instructie in Config.db juist is. Als de installatie-instelling van SQLHOME is gewijzigd, controleert u of de volledige set met SQL-systeemtabellen, inclusief Sysdbs, aan de nieuwe SQLHOME-bestandsindex is overgedragen.

Catalogustabel(len) vergrendeld door een ander:<tabelnaam> [1140]

De bewerking kan niet worden voltooid omdat op de SQL-catalogustabellen vergrendelingen zijn aangebracht. Wacht even en probeer de bewerking opnieuw uit te voeren. Als deze melding vaak wordt weergegeven, stelt u met SET REPROCESS het aantal iteraties hoger in.

Catalogustabellen kunnen alleen worden gelezen:<tabelnaam> [1161]

Alleen degene die de gebruikersnaam SQLDBA heeft, kan catalogustabellen wijzigen in systemen die met een wachtwoord zijn beveiligd.

CHECK OPTION niet mogelijk in huidige visie [1107]

Bij een visie die niet kan worden bijgewerkt, kunt u de clausule WITH CHECK OPTION niet toepassen. Raadpleeg het commando CREATE VIEW in hoofdstuk 6 voor de regels die gelden voor het bijwerken van visies.

Clausule ADD ontbreekt in ALTER TABLE [1036]

De clausule ADD van het commando ALTER TABLE is niet aanwezig, niet juist gespeld of niet op de juiste positie geplaatst.

Clausule BY wordt niet ondersteund en daarom genegeerd

Dit is een waarschuwing en geen foutmelding. In de opdrachten GRANT en REVOKE wordt door dBASE IV SQL geen BY-clausule ondersteund. U hoeft geen actie te ondernemen.

Clausule FOR UPDATE OF ontbreekt in cursordefinitie [1123]

Tijdens de uitvoering van de opdracht UPDATE WHERE CURRENT OF of DELETE WHERE CURRENT OF is een fout opgetreden omdat het commando DECLARE CURSOR waarmee de cursor voor UPDATE is gedefinieerd, geen FOR UPDATE OF-clausule bevatte. Neem in de opdracht DECLARE CURSOR van de cursor die voor UPDATE wordt gebruikt, de clausule FOR UPDATE OF op. Houd er rekening mee dat de clausule ORDER BY niet beschikbaar is wanneer de clausule FOR UPDATE OF is opgenomen.

Clausule GROUP BY ontbreekt [1129]

De clausule SELECT bevat zowel SQL-verzamelfuncties (AVG, COUNT, MAX, MIN of SUM) als kolomnamen. Alle kolomnamen in de clausule SELECT die geen deel uitmaken van de verzamelfuncties, moeten in de clausule GROUP BY worden opgenomen.

Clausule HAVING moet verzamelfuncties bevatten [1116]

Met de clausule HAVING wordt een selectiecriterium opgegeven voor gegroepeerde gegevens. Deze clausule komt meestal na een GROUP BY-clausule. In de HAVING-clausule moet u minstens één van de verzamelfuncties AVG(), COUNT(), MAX(), MIN() en SUM() opgeven.

Clausule INTO niet toegestaan in cursordefinitie **[1127]**
In de opdracht DECLARE CURSOR kan SELECT geen INTO-clausule bevatten. Verwijder de clausule INTO.

Clausule ORDER BY niet toegestaan in CREATE VIEW **[1056]**
In het gedeelte AS SELECT van de opdracht CREATE VIEW is een clausule ORDER BY opgegeven. ORDER BY is echter alleen toegestaan in een volledige SELECT, in combinatie met UNION, of in de opdracht DECLARE CURSOR. Verwijder de clausule ORDER BY. U kunt ORDER BY opgeven als u een selectie uit de visie maakt.

Clausule SAVE TO TEMP niet toegestaan **[1147]**
De clausule SAVE TO TEMP is niet toegestaan in een opdracht DECLARE CURSOR of in een opdracht SELECT waarin de clausule INTO is opgenomen.

Coderingsfout in bestand **[1282]**
In het bestand is een coderingsfout opgetreden. Controleer het gecodeerde bestand. Een ongecodeerde versie van het bestand maakt u door SET ENCRYPTION op OFF in te stellen en het commando LOAD van SQL of COPY TO van dBASE te gebruiken.

Combinatie van afdalend en oplopend sorteren in indexsleutel niet mogelijk **[1183]**
Als u het sleutelwoord DESC hebt opgegeven, moet dit voor alle sleutels in de index worden opgegeven.

Commando kan niet worden uitgevoerd binnen een transactie **[1167]**
De SQL-commando's ALTER, CREATE, DBCHECK, DBDEFINE, DROP, GRANT, REVOKE en RUNSTATS voor het definiëren van gegevens zijn niet toegestaan in transacties. Voer deze commando's buiten transacties uit.

CREATE INDEX/GROUP BY/ORDER BY/SELECT DISTINCT/UNION niet mogelijk in logische kolom **[1182]**
Een index kan niet op basis van een kolom die van het gegevenstype LOGICAL is, worden samengesteld. Alle genoemde bewerkingen hebben betrekking op interne SQL-indexen. Verwijder de kolom van het type LOGICAL uit de opdracht.

Cursor al geopend:<cursornaam>
Met het commando OPEN is in een .prs-programma een cursor opgegeven die al was geopend. Laat het commando OPEN voorafgaan door het commando CLOSE.

Cursor kan niet worden bijgewerkt:<cursornaam> **[1120]**
De opdracht DECLARE CURSOR waarmee de cursor is gedefinieerd, bevat SELECT DISTINCT, UNION, verzamelfuncties, of er is meer dan één tabel of een visie die niet kan worden bijgewerkt, in opgenomen. Gebruik de cursor alleen voor het weergeven van records. Definieer voor de opdrachten DELETE/UPDATE WHERE CURRENT OF en INSERT een andere cursor met DECLARE.

Cursor niet gedefinieerd:<cursornaam> **[1121]**
De desbetreffende cursor is niet met een DECLARE CURSOR-opdracht gedefinieerd. Neem in het .prs-programma de opdracht DECLARE CURSOR op. Zorg ervoor dat de opdracht DECLARE CURSOR aan de eerste opdracht die naar de cursor verwijst, voorafgaat.

Cursor niet geopend:<cursornaam> **[1124]**
Het commando FETCH of CLOSE CURSOR in een .prs-programma kan niet worden uitgevoerd omdat de opgegeven cursor niet met OPEN is geopend. Laat het commando FETCH of CLOSE door het commando OPEN voorafgaan.

Cursordefinitie bevat andere tabelnaam:<cursornaam> **[1126]**
De tabel die in de opdracht DECLARE CURSOR is opgegeven, is niet gelijk aan de tabel die in de opdracht UPDATE/DELETE WHERE CURRENT OF waarin de cursor wordt gebruikt, is opgegeven. Controleer de tabelnaam en de cursornaam zodat u zeker weet dat in de opdrachten DECLARE CURSOR en UPDATE/DELETE WHERE CURRENT OF naar dezelfde tabel wordt verwezen.

Cursornaam eerder vastgelegd:<cursornaam> **[1133]**
De opdracht DECLARE CURSOR bevat een cursornaam die al in een andere DECLARE CURSOR-opdracht in hetzelfde .prg-programma is gebruikt. Kies een nieuwe naam voor de cursor.

Database bestaat niet **[2002]**
Het is mogelijk dat na de compilatie van het programma de gewenste database met DROP is gewist.

Database-naam bestaat al:<database-naam> **[1136]**
De database-naam die in het commando CREATE DATABASE is opgegeven, is al aan een SQL-database gegeven. Gebruik voor de nieuwe database een andere naam.

Database wordt door iemand anders gebruikt - DROP niet mogelijk **[1248]**
DROP DATABASE is niet mogelijk als een andere gebruiker in het netwerk de desbetreffende database heeft geactiveerd.

DBCHECK en RUNSTATS gelden alleen voor basistabellen **[1225]**
Bij het commando DBCHECK of RUNSTATS is een visienaam of een niet-.dbf bestandsnaam opgegeven in plaats van een naam van een basistabel. Met de volgende opdracht kunt u de lijst met niet-visietabellen in de actieve database weergeven: *SELECT Tbname FROM Systabls WHERE Tbtype = "T"*.

DBDEFINE voltooid; fout/waarschuwing gevonden **[2030]**
Ten minste één bestand is niet naar een SQL-tabel geconverteerd vanwege inconsistenties in de SQL-catalogustabellen, vanwege problemen met de bestandsstructuur of omdat deze bestandsnamen al in de systeemcatalogi voorkomen. In de meldingen die aan de foutmelding voorafgaan, wordt aangeduid welke tabellen op de juiste manier met DBDEFINE zijn gedefinieerd en welke niet.

De aanhef van het volgende bestand kan niet worden gelezen:<bestandsnaam>
Tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS is een fout opgetreden omdat de aanhef van een .dbf-bestand niet kon worden gelezen. Als de fout bij DBDEFINE is opgetreden, verwijdert u het bestand met de onbruikbare bestandsaanhef uit de bestandsindex van de database voordat u DBDEFINE opnieuw uitvoert. Als de fout bij DBCHECK of RUNSTATS is opgetreden, moet u de tabel in SQL met DROP wissen voordat u het commando opnieuw kunt uitvoeren.

De volgende bestandsnaam is identiek aan een bestaand synoniem:<tabelnaam>
Deze fout doet zich voor tijdens de uitvoering van DBDEFINE, als één van de te definiëren .dbf-bestanden dezelfde naam heeft als een bestaand SQL-synoniem. Geef het .dbf-bestand een andere naam en voer DBDEFINE opnieuw uit of verwijder het synoniem.

De volgende tabel bestaat al:<tabelnaam>
De in het commando DBDEFINE opgegeven bestandsnaam is al eerder in de SQL-catalogustabellen vastgelegd.

De volgende tabel is niet gevonden in de catalogustabel Systabls:<bestandsnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat er een .dbf-bestand is aangetroffen dat niet voorkomt in de catalogustabel Systabls. Verwijder het .dbf-bestand uit de database-bestandsindex of voer het commando DBDEFINE<bestandsnaam> uit om het .dbf-bestand te definiëren als SQL-tabel. Voer vervolgens DBCHECK of RUNSTATS opnieuw uit.

DELETE-recht niet ontnomen
De gebruiker die dit recht ontneemt, heeft dit nooit toegekend aan de gebruiker aan wie het wordt ontnomen, of het recht is al ontnomen.

DELETE-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht DELETE of heeft dit niet met WITH GRANT OPTION verworven.

DISTINCT moet worden gevolgd door een kolomnaam (niet met logische inhoud) [1097]
Wanneer het sleutelwoord DISTINCT bij de SQL-verzamelfuncties wordt gebruikt, moet dit door een kolomnaam die niet van het gegevenstype LOGICAL is, worden gevolgd.

Dit type gecorreleerde sub-query niet toegestaan [1160]
De SQL-opdracht bevat gecorreleerde, niet gelijke joins, gecorreleerde joins van kolommen van het gegevenstype LOGICAL of gecorreleerde joins met de optie DISTINCT. Verwijder de niet gelijke joins, de joins van kolommen van het gegevenstype LOGICAL of de optie DISTINCT. Maak met SAVE TO TEMP een tijdelijke tabel en gebruik deze voor het ophalen van gegevens voordat u een join uitvoert.

DROP niet mogelijk voor geopende database:<database-naam> [1141]
U hebt geprobeerd met DROP de actieve database te wissen. Geef eerst het commando STOP DATABASE op en daarna DROP DATABASE.

Dubbele waarde in unieke index - afgebroken [2000]
UPDATE of INSERT is niet uitgevoerd omdat hierdoor een waarde in een unieke indexsleutel meer dan eenmaal zou voorkomen. Wijzig de nieuwe waarde zodat deze van alle huidige waarden verschilt. U kunt de index ook met DROP wissen, maar overweeg dit zorgvuldig aangezien met unieke indexen de betrouwbaarheid van de database wordt gegarandeerd.

Dubbelzinnige kolomnaam:<kolomnaam> [1077]
In de huidige opdracht wordt verwezen naar twee kolommen in twee tabellen, maar deze kolommen hebben dezelfde naam. Typ voor dubbelzinnige kolomnamen de tabel- of aliasnaam en een punt:<tabel-/aliasnaam>.<kolomnaam>.

Een visie kan niet worden geïndexeerd:<visienaam> [1117]
In het commando CREATE INDEX is een visienaam opgegeven in plaats van een tabelnaam of synoniem. U kunt een visie niet indexeren.

Eerste argument van clausule LIKE moet een tekenkolom zijn [1090]
De kolomnaam die voorafgaat aan het sleutelwoord LIKE in een WHERE-clausule, duidt geen tekenkolom (CHARACTER) aan. Gebruik LIKE niet bij andere dan tekenkolommen.

Fout in systeemtabel SYSTIMES [1281]
Het bestand Systimes.dbf (dat in netwerken wordt gebruikt voor het registreren van wijzigingen in tabellen) is beschadigd. Iedere gebruiker die de desbetreffende database heeft geopend, moet deze nu sluiten (CLOSE). Vervolgens kopieert u het bestand Systimes.dbf opnieuw vanuit de bestandsindex SQLHOME naar de desbetreffende SQL database-bestandsindex.

Fout tijdens lezen van bestand:<bestandsnaam> **[1277]**
Tijdens het lezen van het bestand is een fout opgetreden.

Fout tijdens openen van bestand:<bestandsnaam> **[1280]**
Tijdens het openen van het bestand is een fout opgetreden.

Fout tijdens schrijven van bestand:<bestandsnaam> **[1278]**
Tijdens het schrijven van het bestand is een fout opgetreden.

Fout tijdens zoeken van bestand:<bestandsnaam> **[1276]**
Tijdens het zoeken van het bestand is een fout opgetreden.

Gebruikersnaam komt meer dan eenmaal voor **[1202]**
De lijst met gebruikersnamen van het commando GRANT of REVOKE bevat dubbele gebruikersnamen. Verwijder de dubbele gebruikersnamen.

Gecorreleerde sub-query niet toegestaan in clausule HAVING **[1160]**
Een deelselectie in een HAVING-clausule kan niet naar dezelfde tabel als de overkoepelende query verwijzen. Splits de query in twee of meer eenvoudigere queries. Hierbij gebruikt u zonodig tabellen die zijn gemaakt met het commando SAVE TO TEMP.

Geen .dbf-bestand in de actieve database **[1224]**
Tijdens de uitvoering van het commando DBDEFINE zijn er (afgezien van de SQL-catalogustabellen) geen .dbf-bestanden aangetroffen. Ga na of de .dbf-bestanden die u als SQL-tabellen wilt definiëren naar de huidige database-bestandsindex zijn gekopieerd, en voer DBDEFINE vervolgens opnieuw uit.

Geen alias gedefinieerd voor automatische samenvoeging:<tabelnaam> **[1142]**
Bij gebruik van een self-join in de clausule WHERE moet voor iedere tabel in de clausule FROM een alias worden gedefinieerd. U kunt niet tweemaal naar dezelfde tabelnaam verwijzen of een synoniem in plaats van een alias gebruiken. Gebruik een FROM-clausule in de vorm FROM<tabelnaam><aliasnaam>. Voorzie de kolomnamen in de clausules WHERE en SELECT vervolgens van de juiste prefixen.

Geen ALTER- of INDEX-rechten voor visies **[1203]**
De rechten ALTER en INDEX kunnen niet worden toegekend of ontnomen, omdat visies niet kunnen worden gewijzigd of geïndexeerd. Voor visies kunt u evenmin het commando GRANT/REVOKE ALL PRIVILEGES geven, omdat onder ALL ook de rechten ALTER en INDEX vallen. Een rechtenlijst voor visies mag de volgende rechten bevatten: DELETE, INSERT, SELECT en UPDATE.

Geen database geopend **[1139]**
Het door u gegeven commando kan alleen worden uitgevoerd als er een database is geopend. Als u SQL hebt geactiveerd terwijl alle databases zijn gesloten, of als u het commando STOP DATABASE hebt gegeven, kan het door u gegeven commando niet worden uitgevoerd. Open een database met START DATABASE voordat u informatie ophaalt.

Geen DBDEFINE voor de tabellen:
De genoemde bestanden zijn niet geconverteerd naar SQL-tabellen. Zie **DBDEFINE voltooid; fout/waarschuwing gevonden.**

Geen gegevens in catalogustabel Sysidxs voor de volgende tabel/index:<tabelnaam en indexnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat er een indexsleutel is aangetroffen met een indexnaam die niet voorkomt in de catalogustabel Sysidxs. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-

bestand en het .mdx-bestand vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Geen gegevens in catalogustabel Syskeys voor de volgende tabel/index/kolom:<tabelnaam, indexnaam, kolomnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat er een indexsleutel is aangetroffen met een indexnaam die niet voorkomt in de catalogustabel Syskeys. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-bestand en het .mdx-bestand vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Geen geldig dBASE/SQL-bestand:<naam database-bestand> [1230]
Tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS is een fout opgetreden omdat uit de gegevens in de bestandsaanhef is gebleken dat het geen dBASE- of SQL-bestand betreft. Controleer of u niet onbedoeld een niet-dBASE- of SQL-bestand met de extensie .dbf hebt gekopieerd. Verwijder het bestand dat de foutmelding veroorzaakte uit de SQL-database, voordat u het commando DBCHECK, DBDEFINE of RUNSTATS opnieuw uitvoert.

Geen records geselecteerd
Een query heeft geen rijen opgeleverd.

Geen rij beschikbaar voor UPDATE of DELETE:<tabel- of visienaam> [1154]
Deze fout doet zich alleen voor in .prs-programma's wanneer UPDATE- en DELETE-bewerkingen onder besturing van SQL-cursors worden uitgevoerd. Ga na of een commando FETCH is gegeven voordat een UPDATE of DELETE wordt uitgevoerd.

Geen toekenning mogelijk wanneer betreffende gebruiker 'PUBLIC' is
Dit is een waarschuwing en geen foutmelding. GRANT OPTION wordt genegeerd omdat de rechten als PUBLIC worden toegekend en verdere toekenning met GRANT dus niet nodig is. U hoeft geen actie te ondernemen.

Gegevens aanwezig in catalogustabel Syscols voor de volgende tabel/kolom:<tabelnaam en kolomnaam>
U hebt met DBDEFINE geprobeerd een tabel te definiëren waarvoor al gegevens in de catalogustabel Syscols aanwezig zijn. Deze foutmelding geeft aan dat er in de catalogustabellen van de bestandsindex van de database ernstige inconsistenties voorkomen. Maak met CREATE een nieuwe bestandsindex voor de database, kopieer alle .dbf- en .mdx-bestanden die niet bij de catalogus horen naar een nieuwe database en geef ten slotte het commando DBDEFINE, of wijzig de kolomnaam met het dBASE-commando MODIFY STRUCTURE voordat u DBDEFINE gebruikt.

Gegevens aanwezig in catalogustabel Sysidxs voor de volgende tabel/index:<tabelnaam en indexnaam>
U hebt met DBDEFINE geprobeerd een index te definiëren waarvoor al gegevens in de catalogustabel Sysidxs aanwezig zijn. Deze foutmelding geeft aan dat er in de catalogustabellen van de bestandsindex van de database ernstige inconsistenties voorkomen. Maak met CREATE een nieuwe bestandsindex voor de database, kopieer alle .dbf- en bijbehorende .mdx-bestanden die niet bij de catalogus horen, naar een nieuwe database en geef ten slotte het commando DBDEFINE, of ken in dBASE een nieuwe naam aan de label toe voordat u DBDEFINE gebruikt.

Gegevens aanwezig in catalogustabel Syskeys voor de volgende tabel/index/kolom:<tabelnaam, indexnaam, kolomnaam>
U hebt met DBDEFINE geprobeerd een indexsleutel te definiëren waarvoor al gegevens in de catalogustabel Syskeys aanwezig zijn. Deze foutmelding geeft aan dat er in de

catalogustabellen van de bestandsindex van de database ernstige inconsistenties voorkomen. Maak met CREATE een nieuwe bestandsindex voor de database, kopieer alle .dbf- en bijbehorende .mdx-bestanden die niet bij de catalogus horen, naar een nieuwe database en geef ten slotte het commando DBDEFINE, of baseer de label op een andere uitdrukking in dBASE voordat u DBDEFINE gebruikt.

Gegevenstype in kolom komt niet overeen met tabel/kolom:<tabel>
Bij het commando DBCHECK of RUNSTATS is een fout opgetreden omdat het gegevenstype dat in de kolom Coltype van de catalogustabel Syscols wordt aangeduid, niet met het gegevenstype van de eigenlijke kolom overeenkomt. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx- of .dbt-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Daarna kopieert u deze bestanden terug naar de oorspronkelijke bestandsindex en definieert u met DBDEFINE<bestandsnaam> de tabel en de indexen opnieuw.

Gegevenstype van kolommen bij UNION-bewerking komt niet overeen [1093]
De overeenkomstige kolommen van SELECT-opdrachten die met het sleutelwoord UNION zijn samengevoegd, zijn niet van hetzelfde gegevenstype. Controleer het gegevenstype van overeenkomstige kolommen. De typen SMALLINT, INTEGER, DECIMAL, NUMERIC en FLOAT worden bij UNION-bewerkingen als hetzelfde type beschouwd maar u moet er wel dezelfde lengte en hetzelfde aantal decimale posities voor opgeven. De overige gegevenstypen moeten exact met elkaar overeenkomen.

Geheel getal noodzakelijk [1032]
Er is een waarde aangetroffen die geen geheel getal is. Vervang deze waarde door een geheel getal.

Geheugenvariabele en dBASE-functie niet toegestaan in clausule SELECT met UNION [1163]
Bij gebruik van UNION moeten alle SELECT-kolommen benoemde kolommen of constanten zijn. Verwijder eventuele geheugenvariabelen, dBASE-functies en reeksverwijzingen uit de SELECT-clausules.

Geheugenvariabele/kolomnaam niet gedefinieerd of ongeldig type geheugenvariabele [2007]
Er is een naam aangetroffen die geen kolomnaam is en ook niet als geheugenvariabele is gedefinieerd, of het gegevenstype van de geheugenvariabele komt niet overeen met het gegevenstype dat in de SQL-opdracht wordt aangegeven. Ga na of een kolomnaam of geheugenvariabele een typefout bevat. Controleer bovendien of de opgegeven kolommen afkomstig zijn uit de tabel waarnaar wordt verwezen. Stel ten slotte vast of er met STORE waarden zijn toegewezen aan de geheugenvariabelen en of de toegewezen waarden compatibel zijn met de gegevenstypen die in de SQL-opdracht zijn vereist.

Gelijkteken ontbreekt [1022]
Na het sleutelwoord SET<kolomnaam> in een UPDATE-opdracht is het gelijkteken niet aanwezig of niet op de juiste positie geplaatst.

Geneste functie niet toegestaan [1098]
Een verzamelfunctie bevat een andere, geneste verzamelfunctie. Verwijder de geneste verzamelfunctie.

GRANT-optie genegeerd voor UPDATE met kolommenlijst
Dit is een waarschuwing en geen foutmelding. Degene aan wie het recht is toegekend, kan het mutatierecht (UPDATE) niet met GRANT aan anderen toekennen omdat het mutatierecht alleen voor bepaalde kolommen is toegekend. U hoeft geen actie te ondernemen. Als het mutatierecht zonder kolommenlijst is toegekend, kan WITH GRANT OPTION wel worden uitgevoerd.

GROUP BY, HAVING, ORDER BY, UNION of FOR UPDATE OF niet toegestaan met INTO **[1128]**
Een SELECT-opdracht in een .prs-programma bevat zowel de clausule INTO als een van de clausules GROUP BY, HAVING, ORDER BY, UNION en FOR UPDATE OF. Een SELECT-opdracht die een INTO-clausule bevat, moet resulteren in één rij en kan geen van deze clausules bevatten.

GROUP BY-kolom(men) niet opgegeven in clausule SELECT **[1094]**
In de clausule GROUP BY is een kolom opgegeven die niet in de clausule SELECT is opgenomen. Neem de kolommen van GROUP BY in de clausule SELECT op. Alle kolommen in de clausule SELECT die niet bij GROUP BY horen, moeten afgeleiden van verzamelfuncties zijn.

Haakjes zijn niet geopend **[1033]**
Een lijst begint niet met het vereiste haakje openen. Raadpleeg de syntaxisbeschrijving van het commando.

Haakjes zijn niet gesloten **[1014]**
U hebt een haakje openen getypt zonder een bijbehorend haakje sluiten. Controleer de ingevoerde haakjes in deelselecties, verzamelfuncties, enzovoort.

Het volgende bestand is niet gevonden in de actieve database:<bestandsnaam>
Tijdens de uitvoering van DBCHECK of RUNSTATS is in de SQL-systeemcatalogi een bestandsnaam aangetroffen die niet beschikbaar is als .dbf-bestand in de database-bestandsindex. Vermoedelijk is het bestand buiten SQL gewist (in dBASE of in DOS).

In clausule SAVE TO TEMP moeten kolom-/veldnamen worden opgegeven **[1149]**
SELECT resulteert in kolommen die van functies of constanten zijn afgeleid. Daarom moeten de kolomnamen van SAVE TO TEMP worden opgegeven. Voeg achter de bestandsnaam in de clausule SAVE TO TEMP een lijst met kolomnamen tussen haakjes toe.

In functie AVG, MAX, MIN of SUM ontbreekt kolomnaam **[1130]**
Voor de SQL-verzamelfuncties AVG(), MAX(), MIN() en SUM() is als argument een kolomnaam vereist.

INDEX-recht niet ontnomen
De gebruiker die dit recht ontneemt, heeft dit nooit toegekend aan de gebruiker aan wie het wordt ontnomen, of het recht is al ontnomen.

INDEX-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht INDEX of heeft dit niet met WITH GRANT OPTION verworven.

Indexnaam bestaat al **[1102]**
De indexnaam in het commando CREATE INDEX is al voor een SQL-index gebruikt. Geef voor de nieuwe index een andere indexnaam op, want elke indexnaam in een SQL-database moet uniek zijn.

INSERT-recht niet ontnomen
De gebruiker die dit recht ontneemt, heeft dit nooit toegekend aan de gebruiker aan wie het wordt ontnomen, of het recht is al ontnomen.

INSERT-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht INSERT of heeft dit niet met WITH GRANT OPTION verworven.

Interne fout nr 1 in SQL-hulpprogramma **[1218]**
Er is een fout opgetreden in de dBASE-routine die gegevens terugzendt naar DBCHECK, DBDEFINE of RUNSTATS.

Interne fout nr 2 in SQL-hulpprogramma **[1219]**
Er is een onjuiste bestandswijzer aangetroffen tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS.

Interne fout nr 3 in SQL-hulpprogramma **[1220]**
Er is een onjuiste structuurwijzer aangetroffen tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS.

Interne fout nr 4 in SQL-hulpprogramma **[1221]**
Er is een onjuiste indexwijzer aangetroffen tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS.

Interne fout nr 5 in SQL-hulpprogramma **[1222]**
Er is een onjuiste kolomwijzer aangetroffen tijdens de uitvoering van DBCHECK, DBDEFINE of RUNSTATS.

Interne SQL-fout nr 1 **[1261]**
Ongedefinieerde verwijzingsnaam.

Interne SQL-fout nr 2 **[1262]**
De tekenreeks in de tabel is te lang. Vereenvoudig de query.

Interne SQL-fout nr 3 **[1263]**
Ongeldig SQL-commando.

Interne SQL-fout nr 4 **[1264]**
Interne overloop van de relatietabel. Vereenvoudig de query.

Interne SQL-fout nr 5 **[1265]**
Een fout in de structuur voor de syntaxiscontrole. Vereenvoudig de query.

Interne SQL-fout nr 6 **[1266]**
Er is een unieke index gebruikt en in meer dan 10 unieke indexen wordt dezelfde kolom voor de unieke sleutel gebruikt. Verwijder enkele unieke indexen.

Interne SQL-fout nr 7 **[1267]**
Een interne fout door een geopend bestand.

Interne SQL-fout nr 8 **[1268]**
Interne overloop van de relatietabel. Vereenvoudig de query.

Interne SQL-fout nr 9 **[1269]**
Er zijn te veel kolommen in tijdelijke relaties gemaakt. Vereenvoudig de query.

Interne SQL-fout nr 10 **[1270]**
Een fout bij het samenvoegen van klassen door de optimalisator. Vereenvoudig de query.

Interne SQL-fout nr 11 **[1271]**
Overloop van de tijdelijke catalogustabel Systabl in de optimalisator. Vereenvoudig de query.

Interne SQL-fout nr 12 **[1272]**
Overloop van de tijdelijke catalogustabel SYSCOLUMNS in de optimalisator. Vereenvoudig de query.

Interne SQL-fout nr 13 **[1273]**
Een fout in de tijdelijke systeemcatalogi van de optimalisator. Vereenvoudig de query.

Interne SQL-fout nr 14 **[1274]**
Een ongeldige wijzer voor het vrijmaken van toegewezen geheugenruimte.

Interne SQL-fout nr 15 **[1275]**
Er is onvoldoende geheugenruimte beschikbaar voor toekenning. Vereenvoudig de queries door een lang commando in verschillende kortere commando's op te splitsen.

Interne SQL-fout nr 16 **[1276]**
Een fout tijdens het zoeken van een bestand.

Interne SQL-fout nr 17 **[1277]**
Een fout tijdens het lezen van een bestand.

Interne SQL-fout nr 18 **[1278]**
Een fout tijdens het schrijven van een bestand

Interne SQL-fout nr 19 **[1279]**
Een ernstige interne tabelfout: overloop in de systeemcatalogi. Vereenvoudig de query.

Interne SQL-fout nr 21 **[1281]**
Een fout in Systimes.

Interne SQL-fout nr 22 **[1282]**
Een fout in de bestandscodering. Controleer de bestandscodering. U kunt SET ENCRYPTION op OFF instellen en het commando LOAD van SQL of COPY TO van dBASE gebruiken om een ongecodeerde versie van het bestand te maken.

Interne SQL-fout nr 24 **[1284]**
Overloop in de query-array.

Interne SQL-fout nr 29 **[1245]**
De ingevoerde dBASE-bronregel is langer dan 1024 bytes. Zoek naar een * in een SELECT-clausule of naar een samengestelde WHERE-operator, met name naar IN, ANY en ALL.

INTO niet mogelijk in definitie van visie **[1106]**
Het sleutelwoord INTO is opgenomen in een CREATE VIEW-commando. Verwijder INTO uit de visiedefinitie.

Kolom kan niet worden bijgewerkt:<kolomnaam> **[1109]**
Bij het commando INSERT of UPDATE is een afgeleide kolom of een kolom in een visie die niet kan worden bijgewerkt, opgegeven. Raadpleeg de bespreking van het commando CREATE VIEW in hoofdstuk 6 voor de regels die gelden voor het bijwerken van kolommen.

Kolom niet gevonden in catalogustabel Syscols voor tabel/kolom:<tabelnaam en kolomnaam>
Bij het commando DBCHECK of RUNSTATS is een fout opgetreden omdat een veld in het .dbf-bestand niet als een kolom in de catalogustabel Syscols werd aangetroffen. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens met DROP de SQL-tabel. Daarna kopieert u de .dbf- en .mdx-bestanden terug naar de oorspronkelijke bestandsindex en definieert u met DBDEFINE<bestandsnaam> de tabel en de indexen opnieuw.

Kolommenlijst voor UPDATE genegeerd bij REVOKE
Dit is een *waarschuwing*, geen fout. Het mutatierecht wordt automatisch voor alle kolommen in een tabel ingetrokken. Vervolgens kunt u het mutatierecht wel met het commando GRANT voor bepaalde kolommen opnieuw toekennen.

Kolomnaam bestaat al:<kolomnaam> **[1081]**
Met een commando waarmee nieuwe kolommen worden benoemd, is één naam tweemaal opgegeven, of is geprobeerd een naam te gebruiken die reeds voor een kolom in de tabel is gebruikt. Kies een andere kolomnaam.

Kolomnaam of -nummer ontbreekt na ORDER BY **[1020]**
In de ORDER BY-lijst kunnen alleen kolomnamen of gehele getallen worden opgegeven. Geef na het sleutelwoord ORDER BY de kolommen op waarop u de resultatentabel wilt ordenen of vervang deze door gehele getallen waarmee de positie van de kolommen in de SELECT-clausule wordt aangeduid.

Kolomnamen in visie moeten worden opgegeven **[1105]**
De clausule SELECT van een opdracht CREATE VIEW bevat kolommen die zijn afgeleid van verzamelfuncties, dBASE IV-functies, constanten, geheugenvariabelen of twee verschillende tabellen. Kolomnamen in een visie kunnen niet zonder meer uit de clausule SELECT worden overgenomen. Voeg aan de sleutelwoorden CREATE VIEW een lijst kolomnamen tussen haakjes toe.

Komma invoeren of haakjes sluiten **[1052]**
U hebt een lijst onjuist opgegeven. Controleer of er zich komma's tussen de onderdelen bevinden en of aan het einde van de lijst een haakje sluiten staat.

Lengte van kolom komt niet overeen met tabel/kolom:<tabelnaam en kolomnaam>
Bij de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat de kolomlengte die wordt aangegeven in de kolom Collen van de catalogustabel Syscols, niet overeenkomt met de werkelijke lengte van de kolom. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens de SQL-tabel met DROP. Kopieer het .dbf-bestand en het .mdx-bestand vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Let op — clausule WHERE ontbreekt in opdracht DELETE
Een waarschuwing ter informatie, die verschijnt als een commando DELETE zonder clausule WHERE wordt opgegeven. Bij uitvoering van het commando worden alle rijen in de tabel gewist. Druk op Esc als u de bewerking wilt annuleren, of op een willekeurige andere toets om alle rijen te wissen.

Let op — clausule WHERE ontbreekt in opdracht UPDATE
Een waarschuwing ter informatie, die verschijnt als een commando UPDATE zonder clausule WHERE wordt opgegeven. Bij uitvoering van het commando worden alle rijen in de tabel bijgewerkt. Druk op Esc als u de bewerking wilt annuleren, of op een willekeurige andere toets om alle rijen bij te werken.

Let op — Cursor<cursornaam> niet gesloten
Een waarschuwing ter informatie. Een geopende cursor is vóór het einde van het programma of de procedure niet door de gebruiker gesloten. Bij het afsluiten van een programma of procedure worden alle cursors die gedurende de bewerking zijn geopend, door SQL gesloten.

Let op — DROP DATABASE verwijdert alle SQL-tabellen
Een waarschuwing ter informatie, die verschijnt als u het commando DROP DATABASE geeft. Als u verder gaat, worden alle SQL-tabellen, zowel de catalogustabellen als de gegevenstabellen, verwijderd. De informatie over de database in de catalogustabel Sysdbs wordt ook verwijderd. Druk op Esc als u de bewerking wilt annuleren, of op een willekeurige andere toets om de database te wissen.

LOAD en UNLOAD DATA niet toegestaan voor visies **[1143]**
De commando's LOAD en UNLOAD DATA kunnen alleen worden gebruikt met SQL-basistabellen of de bijbehorende synoniemen. U kunt op de visie de opdracht SELECT met de clausule SAVE TO TEMP uitvoeren en vervolgens met UNLOAD de gegevens uit de tijdelijke tabel exporteren.

Met GROUP BY gedefinieerde visie kan niet in een samenvoeging worden gebruikt **[1110]**
Een visie waarnaar wordt verwezen in de clausule FROM van een opdracht voor het samenvoegen van visies (en tabellen), is gemaakt met een visie-definitie waarin de clausule GROUP BY is gebruikt. Deze visie kan niet in een join (samenvoeging) worden gebruikt.

Met GROUP BY gedefinieerde visie onbruikbaar in query met clausule GROUP BY **[1111]**
Een opdracht SELECT met een clausule GROUP BY verwijst naar een visie waarvan de definitie een clausule GROUP BY bevat. Verwijder de clausule GROUP BY uit de opdracht SELECT of geef de genoemde visie niet op in deze opdracht.

Naam bestaat al:<objectnaam> **[1080]**
De naam die u hebt opgegeven in het commando CREATE TABLE, CREATE SYNONYM of CREATE VIEW, is eerder al gedefinieerd als naam van een SQL-tabel, een synoniem of een visie. Kies een andere naam voor de tabel, het synoniem of de visie.

Naam, constante of uitdrukking ontbreekt **[1010]**
Er ontbreekt een scalaire waarde. Ga na of in de opdracht een kolomnaam, een constante of een uitdrukking ontbreekt. Controleer bovendien of de opdracht niet te veel komma's bevat. Tussen de afzonderlijke clausules in de opdracht mogen geen komma's worden geplaatst.

Naam is meer dan 10 tekens lang **[1001]**
Kolomnamen en geheugenvariabelen in SQL-opdrachten mogen niet langer dan 10 tekens zijn. (Namen van databases, tabellen, visies, synoniemen en indexen mogen uit ten hoogste acht tekens bestaan.)

Naam ontbreekt **[1012]**
Een objectnaam ontbreekt, staat op de verkeerde plaats of bevat een typefout. Controleer of de database-naam en de namen van tabellen, visies, synoniemen of kolommen correct zijn opgegeven. Controleer bovendien of de opdracht niet te veel komma's bevat. Tussen de afzonderlijke clausules in de opdracht mogen geen komma's worden geplaatst.

Naam van meer dan 8 tekens niet toegestaan **[1156]**
Een opgegeven naam van een database, tabel, index, synoniem of visie bestaat uit meer dan acht tekens. Kies een naam van ten hoogste acht tekens (de naam moet met een letter beginnen en mag verder alleen bestaan uit letters, cijfers en onderstrepingstekens).

Niet-gedefinieerde database-naam:<database-naam> **[1137]**
De genoemde database is niet vastgelegd in de catalogustabel Sysdbs. Controleer of u de juiste database-naam hebt opgegeven of voer CREATE DATABASE uit om de database te definiëren.

Niet-gedefinieerde indexnaam:<indexnaam> **[1088]**
De in het commando DROP INDEX opgegeven index is niet als SQL-index gedefinieerd (de indexnaam is niet vastgelegd in de SQL-catalogustabel SYSIDXS). Voer een SELECT-bewerking op de catalogustabel Sysidxs uit om de juiste naam van de index te achterhalen.

Niet-gedefinieerde kolomnaam:<kolomnaam> **[1076]**
De kolomnaam die deze fout heeft veroorzaakt, komt niet voor in één van de tabellen waarnaar in het commando wordt verwezen. U hebt naar de verkeerde tabel verwezen, een tabel vergeten op te nemen in de clausule FROM, de kolomnaam verkeerd getypt of een niet-bestaande kolomnaam opgegeven.

Niet-gedefinieerd recht in opdracht GRANT of REVOKE **[1055]**
U kunt de volgende rechten verlenen (GRANT) of intrekken (REVOKE): SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER. Ga na of u typefouten hebt gemaakt of een sleutelwoord hebt opgegeven dat niet is toegestaan in het commando GRANT of REVOKE (bijvoorbeeld het sleutelwoord CREATE).

Niet-gedefinieerd symbool **[1004]**
De opdracht bevat een niet-toegestaan teken of u hebt een teken op de verkeerde plaats gebruikt. Ga na of u bijvoorbeeld het teken $ of & hebt gebruikt of in namen of sleutelwoorden tekens hebt opgenomen die als (vergelijkings)operator worden gebruikt.

Niet-gedefinieerd synoniem:<synoniem> **[1089]**
Het in het commando DROP SYNONYM opgegeven synoniem is niet als SQL-synoniem gedefinieerd (het synoniem is niet vastgelegd in de catalogustabel Syssyns). Ga na of u per ongeluk een tabel- of visienaam hebt opgegeven. Voer een SELECT-bewerking op de catalogustabel Syssyns uit om het juiste synoniem te achterhalen.

Niet-gedefinieerde tabelnaam:<tabelnaam> **[1086]**
De tabel, het synoniem of de visie waarom u hebt gevraagd, is niet vastgelegd in de desbetreffende SQL-catalogustabel. U hebt een typefout gemaakt of het commando CREATE (of DBDEFINE) is voor het desbetreffende object nooit uitgevoerd. Ga na of u een kolomnaam of andere objectnaam hebt opgegeven in plaats van de vereiste tabelnaam. Voer een SELECT-bewerking uit op de desbetreffende catalogustabel (Systabls, Syssyns of Sysviews) om een overzicht van de juiste namen te verkrijgen.

Niet-gedefinieerde visienaam:<visienaam> **[1087]**
De in het commando DROP VIEW opgegeven visie is geen SQL-visie (de visienaam is niet vastgelegd in de catalogustabel Sysviews). Voer een SELECT-bewerking uit op de catalogustabel Sysviews om een overzicht van de bestaande visienamen te krijgen.

Niet-numerieke array-subscript **[1162]**
Een array-subscript moet altijd in een geheel getal resulteren. Als u kolomnamen of functies hebt gebruikt, controleert u of het gegevenstype van de uitdrukking numeriek is.

Numerieke waarde te groot **[1144]**
Een numerieke waarde is groter dan de waarde die maximaal in dBASE is toegestaan.

Numerieke waarde te klein **[1153]**
Een numerieke waarde is kleiner dan de kleinste waarde die in dBASE is toegestaan.

Ongeldig aantal decimalen **[1159]**
Bij het commando CREATE TABLE is de lengte van een kolom van het numerieke of decimale gegevenstype onjuist opgegeven. De breedte en schaal van kolommen van het numerieke en decimale gegevenstype worden als (x,y) opgegeven. Beide kolomtypen kunnen als (1,0) worden opgegeven. Als y geen 0 is, mag y niet groter zijn dan x-2. Voor beide typen geldt dat y niet groter mag zijn dan 18.

Ongeldig argument voor COUNT **[1037]**
Het argument voor COUNT moet (*) of (DISTINCT/ALL<kolomnaam>) zijn.

Ongeldig argument voor verzamelfunctie **[1099]**
In de functie AVG(), MAX(), MIN() of SUM() hebt u een asterisk (*) of een niet-toegestaan gegevenstype gebruikt. Kolommen van het logische gegevenstype kunnen niet in SQL-functies worden gebruikt. Het argument van de functies SUM() en AVG() moet een kolom of een uitdrukking zijn die in een numerieke waarde resulteert.

Ongeldig bestandstype **[1050]**
Bij de opdrachten LOAD en UNLOAD kan het bestandstype als SDF, DIF, WKS, SYLK, FW2, RPD, DBASEII of DELIMITED worden opgegeven. Controleer het bestandstype. Als het bestand met LOAD uit een .dbf-bestand wordt geïmporteerd of met UNLOAD naar een .dbf-bestand wordt geëxporteerd, hoeft u het type niet op te geven.

Ongeldig INSERT-onderdeel **[1152]**
Na het sleutelwoord VALUES kunnen de onderdelen van de waardenlijst geen kolomnamen of samengestelde uitdrukkingen met rekenkundige operatoren zijn. Controleer of de waardenlijst alleen de volgende onderdelen bevat: constanten, dBASE-functies, geheugenvariabelen of het sleutelwoord USER.

Ongeldig kolomnummer in clausule ORDER BY **[1084]**
Het kolomnummer is geen geheel getal of het is groter dan het aantal kolommen waarin de clausule SELECT resulteert. Zorg ervoor dat het kolomnummer overeenkomt met de positie van de kolom in de clausule SELECT (de eerste kolom bijvoorbeeld wordt met nummer 1 aangeduid, de tweede kolom met 2, enzovoort).

Ongeldig teken **[1002]**
In objectnamen en kolomnamen zijn alleen letters, cijfers en onderstrepingstekens toegestaan. Getallen mogen alleen bestaan uit cijfers en decimaaltekens. Verwijder de niet-toegestane tekens.

Ongeldig wachtwoord:<gebruikersnaam> **[1283]**
In de bestandsaanhef is een ongeldig wachtwoord ingevoerd.

Ongeldige bestandsnaam **[1049]**
In de LOAD- of UNLOAD-opdracht bevindt zich een ongeldige bestandsnaam. Controleer of de bestandsnaam (en het pad) juist zijn opgegeven. Wanneer het een .dbf-bestandstype betreft, hoeft u de extensie niet op te geven.

Ongeldige constante **[1011]**
In een SQL-opdracht moest u een constante invoeren, maar u hebt geen waarde opgegeven. Controleer of er zich in de waardenlijst haakjes bevinden waartussen niets is ingevoerd.

Ongeldige monadische operator **[1157]**
Een operator is onjuist gecombineerd met een niet-numerieke waarde. Mogelijk hebt u een typefout gemaakt of hebt u een operator geplaatst vóór een kolom van het gegevenstype CHAR, LOGICAL of DATE.

Ongeldige operator in logische vergelijking **[1168]**
Deze melding wordt weergegeven wanneer u andere vergelijkingsoperatoren of sleutelwoorden gebruikt dan de monadische "is gelijk aan"-operator (=) of de monadische "niet gelijk aan"-operatoren (>=, < > of #) in een predikaat waarmee een kolom van het logische type, een geheugenvariabele of reekselement wordt berekend.

Ongeldige rekenkundige uitdrukking **[1135]**
Er wordt een rekenkundige uitdrukking verwacht. Controleer of in de uitdrukking syntaxisfouten voorkomen.

Ongeldige SQL-opdracht **[1059]**
In een stand-alone-versie is het eerste woord niet als SQL-sleutelwoord herkend. Controleer de syntaxis van het commando.

Ongeldige SYSTIME.MEM **[1285]**
De SYSTIME.MEM-tijdsaanduiding in de actieve database-bestandsindex is ongeldig. Ga na of u het bestand Systime.mem onbedoeld hebt overschreven. Maak indien mogelijk een nieuwe kopie van het bestand vanaf de reservekopie van de database.

Ongeldige tabelverwijzing in subselectie met clausule FROM:<tabelnaam> **[1115]**
Naar de tabel waarop het commando INSERT, UPDATE of DELETE wordt uitgevoerd, kan niet nogmaals worden verwezen in de FROM-clausule van een deelselectie.

Ongeldige tekenlengte **[1145]**
De lengte van de desbetreffende tekenreeks is nul of groter dan de maximum lengte in dBASE van 255 tekens.

Ongeldige tekenreeksoperator **[1150]**
Andere operatoren dan plus en min kunnen niet met tekenreeksen worden gecombineerd. Verwijder de ongeldige operator. Controleer bovendien of u niet onbedoeld een naam als tekenreeks hebt gedefinieerd door deze met een aanhalingsteken te beginnen.

Onjuist aantal argumenten in dBASE-functie **[1113]**
Deze melding heeft betrekking op een dBASE-functie die in een SQL-opdracht is gebruikt. Deze foutmelding wordt bijvoorbeeld weergegeven wanneer de functie CTOD() een ongeldige waarde voor de datum bevat, zoals CTOD("99-99-99"). Raadpleeg hoofdstuk 4 voor de juiste syntaxis en het juiste gebruik van de dBASE-functie.

Onjuist aantal INSERT-onderdelen **[1078]**
Het aantal waarden uit de clausule VALUES of de deelselectie komt niet overeen met het aantal kolommen dat in de kolommenlijst is opgegeven (of in de tabel als er geen kolommenlijst is). Controleer of de kolommenlijst de juiste kolommen bevat waarvoor met INSERT gegevens moeten worden tussengevoegd. Controleer of in de VALUES-lijst of in de SELECT-clausule van de deelselectie het juiste aantal waarden is opgegeven.

Onjuist gegevenstype voor argumenten in dBASE-functie **[1112]**
Deze melding heeft betrekking op een dBASE-functie die in een SQL-opdracht is gebruikt. Raadpleeg hoofdstuk 4 voor de juiste gegevenstypen in de dBASE-functie.

Onjuist opgestelde sub-query **[1172]**
Een operator uit een hogere query is in een geneste query geplaatst. Herschrijf de query zonder een sub-query op te nemen (dat wil zeggen: gebruik de operator van de hogere query om het aantal op te halen rijen te beperken).

Onjuiste combinatie van gegevenstypen in uitdrukking **[1131]**
In een uitdrukking zijn kolommen of constanten met onverenigbare gegevenstypen gebruikt. De typen CHAR, LOGICAL en DATE kunnen niet met gegevenstypen die numerieke waarden bevatten en ook niet met elkaar worden gecombineerd. Controleer of de gegevenstypen van de kolommen die in de uitdrukking zijn gebruikt, met de andere kolommen, constanten of functies kunnen worden gecombineerd.

Onjuiste combinatie van gegevenstypen in vergelijking **[1132]**
De gegevenstypen van de kolommen, constanten of uitdrukkingen in een vergelijking komen niet overeen. Controleer de gegevenstypen van de gebruikte kolommen. Controleer of de tekenreeksconstanten tussen aanhalingstekens staan.

Onvoldoende geheugen **[1275]**
De beschikbare geheugenruimte is onvoldoende om de nieuwe gegevens daaraan toe te wijzen. Vereenvoudig queries door lange opdrachten in een aantal kleinere op te delen.

ORDER BY-kolom(men) niet opgegeven in clausule SELECT **[1095]**
In de clausule ORDER BY hebt u een kolom opgegeven die niet voorkomt in de clausule SELECT. Geef de desbetreffende kolomnaam (-namen) ook op in de clausule SELECT.

Pad is te lang **[1044]**
Een padnaam mag niet langer zijn dan 64 tekens.

Positie van kolom in indexsleutel komt niet overeen met tabel/index/kolom:<tabelnaam, indexnaam en kolomnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat de positie van een kolom in een indexsleutel, zoals die is aangegeven in de catalogustabel Syskeys, niet overeenkomt met de werkelijke positie van de kolom in de index. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis de SQL-tabel met DROP. Kopieer het .dbf-bestand en de .mdx-bestanden vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Positie van kolom komt niet overeen met tabel/kolom:<tabel>
Bij het commando DBCHECK of RUNSTATS is een fout opgetreden omdat de gegevens in de catalogustabel Syscols over de positie van een kolom in een tabel niet overeenkomen met de werkelijke positie van de kolom in de tabel. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis vervolgens met DROP de SQL-tabel. Daarna kopieert u het .dbf-bestand en het .mdx-bestand terug naar de oorspronkelijke bestandsindex en definieert u met DBDEFINE<bestandsnaam> de tabel en de indexen opnieuw.

Rechte haken zijn niet gesloten **[1060]**
U hebt een linker rechte haak getypt zonder een bijbehorende rechter haak. Controleer de reeksverwijzingen op de ingevoerde rechte haken. Rechte haken worden behalve in reeksverwijzingen verder niet gebruikt in SQL.

Rij strijdig met definitie van visie — INSERT/UPDATE van rij verworpen **[2005]**
Een UPDATE of INSERT-bewerking met een rij in een visie is niet mogelijk, omdat bij het maken van de visie WITH CHECK OPTION werd opgegeven en de ingevoegde of bijgewerkte rij strijdig is met de definitie van de visie. U kunt de definitie van de visie lezen door de kolom Sqltext uit de catalogustabel Sysviews te selecteren. U kunt INSERT of UPDATE ook uitvoeren voor de onderliggende basistabel, maar de rijen die het resultaat zijn van deze commando's, verschijnen dan niet in de visie.

RUNSTATS voltooid zonder wijziging van catalogus; waarschuwing/fout gevonden **[2009]**
RUNSTATS is niet met succes voltooid, omdat de database inconsistente gegevens bevat. Er is geen enkele wijziging vastgelegd, ook niet voor tabellen waarvoor geen fout of waarschuwing is verschenen. Raadpleeg de foutmeldingen die voorafgingen aan deze melding, voer de voorgestelde oplossing uit en geef het commando RUNSTATS daarna opnieuw.

Scheidingsteken moet één teken of sleutelwoord BLANK zijn **[1048]**
Na de sleutelwoorden DELIMITED WITH in een LOAD- of UNLOAD-voorziening moet u BLANK of één teken als scheidingsteken opgeven.

SELECT is alleen mogelijk met GROUP BY-kolommen of verzamelfuncties [1100]
De SELECT-clausule bevat zowel verzamelfuncties (AVG(), COUNT(), MAX(), MIN() en SUM()) als kolomnamen die niet in verzamelfuncties zijn opgenomen. Alle kolommen die niet in verzamelfuncties zijn opgenomen, moeten in een GROUP BY-clausule worden opgenomen. Voeg aan het commando een GROUP BY-clausule toe. Als de GROUP BY-clausule al onderdeel uitmaakt van het commando, voegt u de ontbrekende kolomnamen van de SELECT-clausule daaraan toe of voegt u een verzamelfunctie aan de kolomnaam toe.

SELECT kan de clausules FOR UPDATE OF en ORDER BY niet beide bevatten [1045]
Een opdracht SELECT of DECLARE CURSOR kan niet zowel de clausule FOR UPDATE OF als ORDER BY bevatten. Als u een cursor die is gedefinieerd in een opdracht DECLARE CURSOR, wilt gebruiken in een opdracht UPDATE WHERE CURRENT OF, is de clausule FOR UPDATE OF vereist. Anders laat u deze clausule weg en geeft u alleen de clausule ORDER BY op.

SELECT-recht niet ontnomen
De gebruiker die dit recht aan een andere gebruiker ontneemt, heeft het recht nooit aan deze gebruiker toegekend, of het recht is de gebruiker al eerder ontnomen.

SELECT-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht SELECT of de bevoegdheid om rechten door te geven (WITH GRANT OPTION).

Slechts één DISTINCT toegestaan in een clausule SELECT [1114]
Het sleutelwoord DISTINCT komt meer dan eenmaal voor in een SELECT-clausule. Verwijder de optie DISTINCT.

Sleutels niet uniek, geen index gemaakt [2006]
In het commando CREATE INDEX is de optie UNIQUE opgegeven, maar de kolom(men) op basis waarvan de index moet worden samengesteld, bevat(ten) dubbele waarden. Voeg een kolom aan de indexsleutel toe of wis (met DELETE) de rijen die duplicaten van sleutelwaarden bevatten. Voer CREATE INDEX met de optie UNIQUE vervolgens opnieuw uit.

Sleutelwoord AND ontbreekt [1018]
Achter het eerste argument in de clausule BETWEEN behoort het sleutelwoord AND te staan. Ga na of het sleutelwoord AND ontbreekt, een typefout bevat of op de verkeerde plaats staat.

Sleutelwoord AS ontbreekt [1028]
Het sleutelwoord AS in het commando CREATE VIEW ontbreekt, bevat een typefout of staat op de verkeerde plaats. AS moet volgen op de visienaam (en de optionele kolommenlijst) en voorafgaan aan de deelselectie.

Sleutelwoord ASC of DESC, komma of haakjes sluiten noodzakelijk [1057]
Het commando CREATE INDEX is niet compleet. Voeg desgewenst de optie ASC of DESC toe aan de kolom(men) in de kolommenlijst (gebruik komma's om de namen van elkaar te scheiden) en sluit de lijst af met een haakje sluiten.

Sleutelwoord ASC of DESC, komma of puntkomma ontbreekt [1058]
De clausule ORDER BY is niet compleet. Achter de kolom(men) in de clausule ORDER BY kunt u desgewenst een sorteervolgorde opgeven (ASC of DESC, ASC is de standaardwaarde). De opdracht moet worden afgesloten met een puntkomma (;).

Sleutelwoord BETWEEN, LIKE of IN ontbreekt
U hebt een WHERE-clausule niet correct opgegeven. Ga na wat de juiste syntaxis van de door u gewenste vorm van de WHERE-clausule is.

Sleutelwoord BY ontbreekt **[1016]**
Het sleutelwoord ORDER moet worden gevolgd door het sleutelwoord BY. Ga na of BY ontbreekt, een typefout bevat of op de verkeerde plaats staat.

Sleutelwoord CHECK ontbreekt **[1038]**
Het sleutelwoord CHECK in de clausule WITH CHECK OPTION van het commando CREATE VIEW ontbreekt, bevat een typefout of staat op de verkeerde plaats.

Sleutelwoord CURSOR ontbreekt **[1041]**
Het sleutelwoord CURSOR van de opdracht DECLARE CURSOR in een .prs-programma ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord DATA ontbreekt **[1040]**
In de opdracht LOAD of UNLOAD ontbreekt het sleutelwoord DATA. Ga na of LOAD of UNLOAD direct wordt gevolgd door het sleutelwoord DATA en of u geen typefouten hebt gemaakt.

Sleutelwoord DATABASE ontbreekt **[1051]**
Het sleutelwoord DATABASE in het commando START, STOP of SHOW DATABASE ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord DATABASE, TABLE, INDEX, SYNONYM of VIEW ontbreekt **[1027]**
Achter het sleutelwoord CREATE is een sleutelwoord niet of niet juist opgegeven. Voeg het sleutelwoord DATABASE, TABLE, INDEX, SYNONYM of VIEW alsnog toe aan CREATE.

Sleutelwoord DATABASE, TABLE, INDEX, SYNONYM of VIEW ontbreekt **[1031]**
Achter het sleutelwoord DROP is een sleutelwoord niet of niet juist opgegeven. Voeg het sleutelwoord DATABASE, TABLE, INDEX, SYNONYM of VIEW alsnog toe aan DROP.

Sleutelwoord FOR ontbreekt **[1030]**
Het sleutelwoord FOR in het commando CREATE SYNONYM of DECLARE CURSOR ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord FROM ontbreekt **[1015]**
Het sleutelwoord FROM in het commando LOAD, UNLOAD, SELECT, REVOKE, of DELETE ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord GRANT ontbreekt **[1054]**
Het sleutelwoord GRANT in de clausule WITH GRANT OPTION van een GRANT-commando ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord INDEX ontbreekt **[1017]**
De sleutelwoorden CREATE UNIQUE moeten worden gevolgd door het woord INDEX. Ga na of het sleutelwoord INDEX ontbreekt, op de verkeerde plaats staat of een typefout bevat.

Sleutelwoord INTO ontbreekt **[1023]**
Het sleutelwoord INTO in de opdracht FETCH, INSERT of LOAD DATA ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord niet toegestaan in interactieve modus **[1118]**
Een sleutelwoord dat u hebt gebruikt in een interactieve SQL-sessie, mag alleen in .prs-programma's worden gebruikt. Alle sleutelwoorden die verband houden met SQL-cursors, zijn alleen in programma's mogelijk. Vervang het commando door een interactief SQL-commando.

Sleutelwoord OF ontbreekt **[1043]**
Het sleutelwoord OF in de clausule FOR UPDATE OF of WHERE CURRENT OF ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord ON ontbreekt **[1026]**
Het sleutelwoord ON in het commando CREATE INDEX, GRANT of REVOKE ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord OPTION ontbreekt **[1039]**
Het sleutelwoord OPTION in de clausule WITH GRANT OPTION van het commando GRANT of in de clausule WITH CHECK OPTION van de opdracht CREATE VIEW ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord SELECT ontbreekt **[1029]**
Het sleutelwoord SELECT achter het sleutelwoord UNION, of achter het sleutelwoord AS in het commando CREATE VIEW ontbreekt, staat op de verkeerde plaats of bevat een typefout. Het ontbrekende sleutelwoord SELECT is het eerste woord in de deelselectie die in deze opdrachten moet worden opgegeven.

Sleutelwoord SELECT ontbreekt in de opdracht DECLARE CURSOR **[1046]**
Het sleutelwoord SELECT in de opdracht DECLARE CURSOR ontbreekt, staat op de verkeerde plaats of bevat een typefout. Een SELECT-opdracht moet worden opgegeven achter het sleutelwoord FOR.

Sleutelwoord SET ontbreekt **[1021]**
Het sleutelwoord SET in de opdracht UPDATE ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord TABLE ontbreekt **[1035]**
Het sleutelwoord TABLE in het commando ALTER TABLE, LOAD of UNLOAD ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord TEMP ontbreekt **[1053]**
Het sleutelwoord TEMP in de clausule SAVE TO TEMP van de opdracht SELECT ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord TO ontbreekt **[1025]**
Het sleutelwoord TO in het commando GRANT of UNLOAD, of in de clausule SAVE TO TEMP van het commando SELECT ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord UPDATE ontbreekt **[1042]**
Het sleutelwoord UPDATE in de clausule FOR UPDATE OF van de opdracht DECLARE CURSOR ontbreekt, staat op de verkeerde plaats of bevat een typefout.

Sleutelwoord VALUES of SELECT ontbreekt **[1024]**
De opdracht INSERT moet een VALUES-clausule of een deelselectie bevatten. Controleer of u één van de sleutelwoorden hebt opgegeven en of dit sleutelwoord niet op de verkeerde plaats staat of een typefout bevat.

Sleutelwoord van gegevenstype ontbreekt **[1033]**
In het commando CREATE TABLE of ALTER TABLE moet het gegevenstype van een kolom na de kolomnaam worden opgegeven. De gegevenstypen van SQL zijn: SMALLINT, INTEGER, DECIMAL, NUMERIC, FLOAT, CHAR, LOGICAL en DATE. Als het gegevenstype reeds is opgegeven, controleert u of het geen typefouten bevat en of het sleutelwoord of de haakjes zich op de juiste plaats bevindt.

Sleutelwoord WITH ontbreekt **[1047]**
Het sleutelwoord WITH in het commando LOAD of UNLOAD ontbreekt, staat op de verkeerde plaats of bevat een typefout, terwijl er wel scheidingstekens zijn opgegeven. Bij het importeren (LOAD) of exporteren (UNLOAD) van gegevens uit dBASE-bestanden zijn de clausules TYPE en WITH overigens niet vereist.

SQLHOME niet ingesteld in CONFIG.DB **[1250]**
Wanneer u het commando SET SQL ON geeft (de regel SQL = ON in het bestand Config.db opneemt), moet de bestandsindex met de SQL-bestanden in het bestand Config.db zijn opgenomen. Neem de regel SQLHOME =<pad> in Config.db op en start dBASE IV opnieuw.

SQL-index kan niet meer dan 10 kolommen bevatten **[1173]**
U hebt een index gemaakt die op meer dan 10 kolommen is gebaseerd.

SQL-syntaxisfout **[1034]**
Er ontbreekt een naam, sleutelwoord, operator of puntkomma of u hebt een typefout gemaakt bij een van deze elementen. Mogelijk hebt u ook een clausule UNION in de definitie van een visie gebruikt. Controleer of u het SQL-commando met een puntkomma hebt afgesloten en de correcte syntaxis hebt gebruikt. Verwijder desgewenst de clausule UNION uit de definitie van de visie.

SQL-systeemcatalogus kan niet worden geopend:<SQL-catalogusbestand> **[1247]**
De systeemcatalogus ontbreekt of is beschadigd. Zorg ervoor dat het bestand kan worden geopend.

Sub-bestandsindex voor nieuwe database kan niet worden gemaakt **[1155]**
Het pad of station in uw opdracht bestaat niet, of u hebt geprobeerd een database te maken dat hetzelfde pad heeft als SQLHOME. Controleer de naam van het pad en de database.

Sub-query heeft in meer dan één waarde geresulteerd **[2003]**
Een sub-query die volgt op een rekenkundige vergelijkingsoperator in een clausule WHERE, mag niet in meer dan één waarde resulteren. Dit is wel gebeurd, dus heeft zich een fout voorgedaan. U moet de sub-query laten voorafgaan door het sleutelwoord ANY of ALL of in plaats van de operator het sleutelwoord IN in de clausule WHERE gebruiken.

Sub-query heeft niet in een waarde geresulteerd **[2011]**
De sub-query in een clausule WHERE of HAVING heeft geen waarden opgeleverd. (Alleen het predikaat WHERE EXISTS mag een sub-query bevatten die niet in een waarde resulteert.) Wijzig het selectiecriterium in de sub-query. U kunt de query afzonderlijk testen om te controleren of deze in de juiste waarden resulteert.

Systeemtabel onvolledig voor tabel:<tabelnaam> **[2004]**
De catalogustabel Systabls bevat geen informatie over de genoemde tabel. De tabel is gewist of bestaat niet.

Tabel bestaat al **[1216]**
De bestandsnaam die volgt op het sleutelwoord DBDEFINE is al eerder als SQL-tabel gedefinieerd en kan daarom niet opnieuw in de SQL-catalogustabellen worden vastgelegd. Als u deze tabel opnieuw moet definiëren (vanwege DBCHECK-fouten of om een andere reden), kopieert u het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wist u de SQL-tabel met DROP. Vervolgens kopieert u het .dbf-bestand en het .mdx-bestand terug naar de oorspronkelijke database-bestandsindex en voert u DBDEFINE<bestandsnaam> uit.

Tabel komt niet voor in huidige opdracht:<tabelnaam> **[1075]**
De opdracht bevat een verwijzing naar een kolom, die wordt voorafgegaan door een tabelnaam, maar deze tabelnaam is niet opgegeven in een FROM-clausule. Voeg de ontbrekende tabelnaam toe aan de clausule FROM.

Tabel niet gevonden bij SQL-catalogustabellen **[1233]**
De in het commando DBCHECK of RUNSTATS opgegeven tabel komt niet voor in de SQL-catalogustabellen. U kunt de juiste tabelnaam achterhalen door middel van een SELECT-bewerking met de catalogustabel Systabls.

Tekenreeks te lang **[1145]**
De ingevoerde tekenreeks is langer dan de maximale lengte voor tekengegevens in dBASE. Voer een kortere reeks in.

Te veel indexen voor tabel **[1170]**
U kunt niet meer dan 47 indexen voor een tabel maken. Verwijder indexen die u niet meer nodig hebt.

Te veel kolommen in tabel **[1166]**
Dit bericht verschijnt als u probeert met CREATE TABLE of ALTER TABLE een tabel met meer dan 255 kolommen te maken.

Te veel unieke indexen **[1266]**
U hebt een unieke index in gebruik en meer dan tien unieke indexen zijn gebaseerd op dezelfde kolom als deel van een unieke indexsleutel. U moet een aantal van de unieke indexen wissen.

Te veel waarden opgegeven na INSERT **[1085]**
De opgegeven kolommenlijst bevat meer kolommen dan er in de tabel voorkomen. Controleer de kolommenlijst op dubbel opgegeven kolommen en verwijder deze.

Te veel werkgebieden geopend **[2008]**
Het aantal geopende werkgebieden is groter dan tien. Er wordt naar meer dan tien tabellen verwezen in één SQL-tabel of niet alle werkgebieden waren gesloten toen u het commando SET SQL ON gaf. Verwijder enkele tabelverwijzingen in de clausule FROM, maak de SQL-opdracht minder complex of ga terug naar de dBASE-modus en sluit de geopende werkgebieden.

Toegekende rechten onvoldoende **[1204]**
Voor de bewerking die u op de opgegeven tabel of visie wilt uitvoeren, beschikt u niet over juiste rechten. Verzoek degene die de tabel of visie heeft gemaakt, u met GRANT de vereiste rechten toe te kennen. Iedere gebruiker waaraan de rechten met WITH GRANT OPTION zijn toegekend, kan deze met GRANT ook aan u toekennen.

Toegekende rechten onvoldoende voor CREATE VIEW **[1205]**
U kunt met CREATE pas een visie maken wanneer u voor elke tabel waarop de visie is gebaseerd, het recht SELECT hebt. Verwijder uit de definitie van de visie de tabellen waarvoor u dat recht niet hebt. U kunt ook een gebruiker verzoeken met GRANT u het recht SELECT voor de desbetreffende tabel toe te kennen.

Tweede argument van clausule LIKE moet een tekenreeks zijn **[1091]**
LIKE moet worden gevolgd door een tekenreeks, het sleutelwoord USER of een geheugenvariabele van het tekentype. Functies en kolomnamen zijn niet toegestaan.

U kunt uzelf geen toegangsrechten toekennen of ontnemen **[1201]**
In de TO- of FROM-clausule van het commando GRANT of REVOKE geeft u uw eigen gebruikersnaam op. U kunt met GRANT en REVOKE alleen toegangsrechten aan andere gebruikers toekennen en ontnemen. Zorg ervoor dat uw eigen gebruikersnaam niet voorkomt in de lijst met gebruikersnamen die in uw opdracht is opgenomen. Denk eraan dat u zich met uw eigen gebruikersnaam aanmeldt.

Uitdrukking niet toegestaan in clausule GROUP BY/ORDER BY **[1201]**
In GROUP BY-lijsten zijn alleen kolomnamen toegestaan. In ORDER BY-lijsten zijn alleen kolomnamen of gehele getallen toegestaan. Verwijder alle uitdrukkingen uit de clausule.

UNION is niet toegestaan in de definitie van een visie
U hebt het sleutelwoord UNION opgegeven in het commando CREATE VIEW. Bij het maken van een visie kunt u wel kolommen uit meer dan één tabel selecteren, maar een UNION-bewerking is niet toegestaan.

UPDATE-kolom(men) niet opgenomen in cursordefinitie:<kolomnaam> **[1122]**
Met de opdracht UPDATE WHERE CURRENT OF in een .prs-programma worden kolommen bijgewerkt die niet zijn genoemd in de kolommenlijst in de clausule FOR UPDATE OF van de bijbehorende opdracht DECLARE CURSOR. Geef alle bij te werken kolommen op in de kolommenlijst van de clausule FOR UPDATE OF van de opdracht DECLARE CURSOR.

UPDATE-recht niet ontnomen
De gebruiker die dit recht aan een andere gebruiker ontneemt, heeft het recht nooit aan deze gebruiker toegekend, of het recht is de gebruiker al eerder ontnomen.

UPDATE-recht niet toegekend
Degene die het recht toekent, beschikt niet over het recht UPDATE of de bevoegdheid om rechten door te geven (WITH GRANT OPTION).

Vergelijkingsoperator of sleutelwoord ontbreekt **[1134]**
Na de eerste kolomnaam of constante in een WHERE-clausule moet een vergelijkingsoperator of een sleutelwoord worden opgegeven. De vergelijkingsoperatoren zijn: =, >, <, >=, <=, < >,<=,<< of<>. De vergelijkingssleutelwoorden zijn: NOT, LIKE, IN en BETWEEN.

Verzamelfunctie niet toegestaan in clausule WHERE **[1103]**
De SQL-verzamelfuncties AVG(), COUNT(), MAX(), MIN() en SUM() mogen niet in de clausule WHERE worden gebruikt (behalve als onderdeel van de clausule SELECT in een deelselectie). U kunt de bewerking opnieuw definiëren en daarin een deelselectie maken. U kunt bijvoorbeeld *SELECT * FROM Staf WHERE Salaris > AVG(Salaris)* vervangen door *SELECT * FROM Staf WHERE Salaris > (SELECT AVG(Salaris) FROM Staf).*

Visie kan niet worden bijgewerkt:<visienaam> **[1108]**
U hebt geprobeerd een DELETE-bewerking uit te voeren in een visie die niet kan worden bijgewerkt. Visies met een definitie waarin naar meer dan één tabel wordt verwezen of die GROUP BY, SELECT DISTINCT of verzamelfuncties bevatten, kunnen niet worden bijgewerkt.

Volgorde van indexkolom komt niet overeen met tabel/index/kolom:<tabelnaam, indexnaam en kolomnaam>
Tijdens de uitvoering van het commando DBCHECK of RUNSTATS heeft zich een fout voorgedaan, omdat de sorteervolgorde (ASC of DESC) die in de catalogustabel Syskeys wordt aangegeven, niet overeenkomt met de werkelijke sorteervolgorde van de indexkolom. Kopieer eerst het .dbf-bestand en het bijbehorende .mdx-bestand naar een andere bestandsindex en wis de SQL-tabel met DROP. Kopieer het .dbf-bestand en de .mdx-bestanden vervolgens terug naar de oorspronkelijke bestandsindex en voer DBDEFINE<bestandsnaam> uit om de tabel en indexen opnieuw te definiëren.

Waarde buiten bereik **[1146]**
Een waarde met een zwevend decimaalteken die groter dan 10 +38 of kleiner dan 10 -38 is, kan niet worden ingevoerd.

Waarde moet overeenkomen met gegevenstype kolom **[1079]**
Een UPDATE- of INSERT-waarde komt niet overeen met het gegevenstype van de kolom waarin de waarde wordt ingevoerd. Controleer het gegevenstype van de kolom en ga na of datums in de juiste indeling zijn ingevoerd: CTOD(" - - ") of { - - }. Controleer of tekenreeksen tussen aanhalingstekens zijn geplaatst.

Waarde overschrijdt kolombreedte **[1158]**
Een waarde bestaat uit meer tekens dan er in de kolom passen waarin de waarde wordt ingevoerd. Met een query uit de tabel Syscols kunt u de breedte van de desbetreffende kolom achterhalen.

Zet bewerking stop met ESC, of druk op een toets om door te gaan
Een beveiligings-prompt die verschijnt in combinatie met waarschuwingen bij bepaalde bewerkingen (bij UPDATE of DELETE zonder clausule WHERE en bij het commando DROP DATABASE).

Specificaties van dBASE IV

B

In deze bijlage wordt een overzicht gegeven van de specificaties van de bestanden en bewerkingen van dBASE IV.

Database-bestand

Aantal records: 1 miljard
Aantal bytes: 2 miljard
Recordgrootte, in .dbf: 4.000 bytes
Aantal velden: 255

Indexbestand

Aantal indexen per samengesteld indexbestand: 47
Blokgrootte: 16.384 bytes (standaard: 2.048 bytes)

Veldgrootten

Tekenvelden: 254 bytes
Datumvelden: 8 bytes
Logische velden: 1 byte
Velden van het type N: 20 cijfers
Velden van het type Z: 20 cijfers
Tekens in een veldnaam: 10
Memovelden: 64Kb

Arrays

Dimensies: 2
Totale grootte (rijen x kolommen): 1170

Multi-user procedures

Vergrendelingen — maximum aantal vergrendelingen: 50 (bestanden en records)
Opnieuw verwerken — maximum aantal tellingen: 32.000
Regeneratie — maximum aantal seconden: 3.600

Bestandsbewerkingen

(Houd er rekening mee dat het aantal bestanden dat gelijktijdig kan worden geopend, beperkt kan zijn door de instellingen in het bestand Config.sys en Config.db en door de beschikbare geheugenruimte.)

Geopende bestanden van alle typen: 99
Geopende database-bestanden: 10
Geopende memobestanden per actieve database: 1
Geopende indexbestanden per actieve database: 11 (10 maal .ndx, 1 maal .mdx)
Geopende indelingsbestanden per actieve database: 1
Geopende procedurebestanden per uitvoering: 1

Numerieke nauwkeurigheid

(Houd er rekening mee dat het decimaalteken bij de bepaling van de nauwkeurigheid niet als cijfer meetelt.)

Getallen van het type Z:

15,9 cijfers
Grootste getal: 0,9 x E+308
Kleinste getal: 0,1 x E-307

Getallen van het type N:

10 tot 20 cijfers op basis van de instelling in SET PRECISION TO <Nuitdr>
Grootste getal: 0,9 x E+308
Kleinste getal: 0,1 x E-307

Geheugenvariabelen

(Houd er rekening mee dat door de beschikbare geheugenruimte het maximum aantal kleiner kan zijn.)

Standaardaantal: 500 variabelen
In Config.db kan dit aantal worden gewijzigd in een maximum van 15.000 variabelen.

Run-time symbolen

(Houd er rekening mee dat door de beschikbare geheugenruimte het maximum aantal kleiner kan zijn.)

Standaardaantal: 500
In Config.db kan dit aantal worden gewijzigd in een maximum van 15.000 symbolen.

Compilatiefasesymbolen

(Houd er rekening mee dat door de beschikbare geheugenruimte het maximum aantal kleiner kan zijn.)

Standaardaantal: 500
In Config.db kan dit aantal worden gewijzigd in een maximum van 5.000 symbolen.

Capaciteit

Hieronder ziet u een lijst met de capaciteit die u voor de verschillende handelingen in dBASE IV, zoals bewerkingen en het schrijven van rapporten, tot uw beschikking hebt.

(Houd er rekening mee dat door de beschikbare geheugenruimte de capaciteit kan verminderen.)

Tekstverwerker met regelovergang

Aantal regels: 32.000
Regellengte:

MODIFY COMMAND (.PRG)	1.024
MODIFY FILE	1.024
MEMO FIELDS	1.024
SQL (F9)	1.024
HISTORY	1.024
REPORT	255

Formulieren

Aantal rijen: 32.767
Breedte: 80 tekens

Rapporten

Breedte: 255 tekens
Aantal databases: 9
Aantal indexen per database: 10
Aantal relaties: 9
Aantal rapporten per database: onbeperkt
Aantal pagina's: 32.767
Aantal geneste groepzones: 44
Aantal velden: afhankelijk van geheugenruimte
Aantal exemplaren: 32.767

Etiketten

Breedte: 255 tekens
Lengte: 255 regels
Aantal etiketten in de breedte: 15
Aantal exemplaren: 32.767

QBE

Samengevoegde bestanden: 8

SQL

Tabellen in een join: afhankelijk van beschikbare geheugenruimte
Aantal cursors: 10
Aantal indexen per tabel: 47
Lengte SQL-opdracht: 1.024 tekens

Applicatiegenerator

Aantal objecten op werkgebied: afhankelijk van complexiteit van objecten en beschikbare geheugenruimte

Grootte van editor: 4K

Diversen

Lengte commandoregel: 255 bytes vanaf commandostip, 1.024 bytes in bewerkvenster
Maximumniveau waarop besturingscommando's kunnen worden genest: met de DO-parameter in Config.db ingesteld
Procedures per programma-/procedurebestand: 963
Maximumgrootte per procedure: 65.520 bytes code
Maximum aantal actieve procedures: afhankelijk van geheugenruimte
Sorteerniveaus voor gelijktijdige sortering: 16
Maximum aantal GET-commando's in een indelingsbestand: 2.000
Lengte afgedrukte pagina: 32.767 regels
Aantal macro's: 35
Aantal binaire bestanden dat kan worden geladen: 16
Aantal printerstuurprogramma's: 4 in configuratie opgenomen
Aantal lettertypen: 5 per printerstuurprogramma
Aantal werkgebieden: 10
Aantal programmeerbare functietoetsen: 29
Aantal herhalingsaanroepen: 16
Aantal vensters: 20

OPMERKING
Een actieve procedure is een procedure die met RETURN kan worden aangeroepen of die zich in het actieve procedurebestand of programmabestand bevindt.

Voorbeeldbestanden

Deze bijlage bevat beschrijvingen van de database-bestanden die in het *Naslaghandboek* als voorbeeld worden gebruikt, en van de daaraan gekoppelde indexbestanden. Deze bestanden zijn ook te vinden op de voorbeelddiskettes.

Een aantal voorbeelden in het *Naslaghandboek* zijn afkomstig uit Menus.prg, een programma voor het weergeven van horizontale balkmenu's en pop-up menu's. In deze bijlage is de code van Menus.prg volledig uitgeschreven.

Afnemers.dbf

Structuur van database: AFNEMERS.DBF
Aantal records: 8
Laatst gewijzigd op : 17-10-88

Num	Veldnaam	Type	Breed	Dec	Index
1	KLANT_NR	Teken	6		J
2	KLANTNAAM	Teken	30		J
3	ACHTERNAAM	Teken	15		N
4	VOORNAAM	Teken	15		N
5	ADRES	Teken	30		N
6	WOONPLAATS	Teken	20		N
7	PROVINCIE	Teken	2		N
8	POSTCODE	Teken	7		N
9	TELEFOON	Teken	13		N
10	OVERZICHT	Memo	10		N
** Totaal **			152		

Recordnr.	KLANT_NR	KLANTNAAM	ACHTERNAAM	VOORNAAM
1	A00001	WIJSMAN & ZONEN BV	Wijsman	Fred
2	L00001	BALJUW & CO	Baljuw	Sandra
3	C00001	L. G. BLOEM & PARTNERS	Martijn	Rinus
4	L00002	SAUER & LANGWEER	Peters	Kim
5	A00005	SMITS & SMITS	Yzinga	Ger
6	C00002	TIEMESE & KEEFMAN BV	Tiemese	Gemma
7	B12000	VOLLEBREGT IMPORT	Mastenbroek	Floor
8	A10025	PUNTER PROMOTIONS	Beckman	Rien

Het onderstaande is het vervolg van het bestand op de vorige pagina:

ADRES	WOONPLAATS	PROVINCIE	POSTCODE	TELEFOON	OVERZICHT
3232 Marktstraat 32	Nieuw Vennep	NH	2151 XT	02526-7474	MEMO
5132 Leuvensedreef 512	Leiden	ZH	2316 TV	071-338900	MEMO
4818 Aalbeslaan 48	Sint Oedenrode	NB	5491 FC	04138-323232	MEMO
12300 Eikenstraat 123	Damwoude	FR	9104 AB	05111-56033	MEMO
7500 Sint Pieterspoort 75	Loosdrecht	UT	1231 VN	02158-4300	MEMO
310-2090 Corantijnstraat 31	Valkenburg	L	6300 UC	04406-557644	MEMO
8506 Havenboulevard 85	Ten Boer	GR	9724 EK	05902-22677	MEMO
332 Mierikswortelweg 33A	Pannerden	GL	6911 BK	08366-94200	MEMO

In het volgende gedeelte vindt u de informatie die is opgenomen in het memobestand dat bij Afnemers.dbf hoort:

```
KLANT:     Recordnr.      1

85-200     02-08-85
     C-300-400 BOEKENKAST                    535,00     1

KLANT:     Recordnr.      2

86-245     22-09-86
     C-700-2020 DOSSIERKAST,2 LADEN          100,00     2
     C-700-4030 DOSSIERKAST,4 LADEN          150,00     2
86-303     06-12-87
     C-222-1001 STOEL,BUREAU                1750,00     1
87-109     09-03-87
     C-400-2060 TAFEL,BIJZET                 250,00     1
     C-500-6050 LAMP,STAAND                  165,00     1
87-112     20-03-87
     C-222-3020 STOEL,KANTINE                350,00     2

KLANT: Recordnr.      3

86-155 31-06-86
     C-600-5050 BUREAU,SECRETARESSE         1100,00     1
     C-222-3010 STOEL,KANTINE                500,00     1
     C-400-2080 TAFEL,BIJZET                 250,00     1
86-248 28-09-86
C-600-5050 BUREAU,SECRETARESSE              1100,00     1
     C-222-3010 STOEL,KANTINE                500,00     1
     C-700-2020 DOSSIERKAST,2 LADEN          100,00     1
     C-500-6050 LAMP,STAAND                 1100,00     1
86-312 17-12-86
     C-400-5000 TAFEL,KOFFIE                 875,00     1
87-106 10-02-87
     C-111-8050 SOFA,8 POTEN                1200,00     1
87-108 23-02-87
     C-222-1000 STOEL,BUREAU                1250,00     1

KLANT: Recordnr.      4

Geen gegevens
```

```
KLANT: Recordnr.          5

 85-187 07-06-85
      C-111-6045 SOFA,8 POTEN          1325,00   1
 87-113 24-03-87
      C-300-2020 BOEKENKAST             125,00   1

KLANT: Recordnr.          6

 87-107 12-02-87
      C-222-1000 STOEL,BUREAU          1250,00   1
 87-110 09-03-87
      C-700-2020 DOSSIERKAST,2 LADEN     75,00   1
      C-700-4020 DOSSIERKAST,4 LADEN    100,00   1

KLANT: Recordnr.          7

 Geen gegevens

 KLANT: Recordnr.         8

 87-105 03-02-87
      C-111-6010 SOFA,6 POTEN          1200,00   1
      C-111-6015 SOFA,6 POTEN           650,00   1
```

Hier ziet u een overzicht van de indexsleutels die voor het bestand Afnemers.dbf worden gebruikt:

```
Hoofdindex-bestand: KLNTNAAM.NDX   Sleutel: Achternaam+Voornaam
Bijbehorend MDX-bestand:      AFNEMERS.MDX
                Indexlabel:   KLANTNAAM          Sleutel: KLANTNAAM
                Indexlabel:   KLANT_NR           Sleutel: KLANT_NR
```

Transact.dbf

```
Structuur van database: TRANSACT.DBF
Aantal records:12
 Laatst gewijzigd op:17-10-88
     Num   Veldnaam      Type       Breed      Dec      Index
       1   KLANT_NR     Teken        6                   J
       2   BESTEL_NR    Teken        6                   J
       3   DAT_TRANS    Datum        8                   N
       4   FACTUUR      Logisch      1                   N
       5   TOTAAL_REK   Numeriek     8          2        N
 ** Totaal **                       30
```

Recordnr.	KLANT_NR	BESTEL_NR	DAT_TRANS	FACTUUR	TOTAAL_REK
1	A10025	87-105	03-02-87	.T.	1850,00
2	C00001	87-106	10-02-87	.T.	1200,00
3	C00002	87-107	12-02-87	.T.	1250,00
4	C00001	87-108	23-02-87	.T.	1250,00
5	L00001	87-109	09-03-87	.T.	415,00
6	C00002	87-110	09-03-87	.T.	175,00
7	L00002	87-111	11-03-87	.F.	1000,00
8	L00001	87-112	20-03-87	.T.	700,00
9	A00005	87-113	24-03-87	.T.	125,00
10	B12000	87-114	30-03-87	.F.	450,00
11	C00001	87-115	01-04-87	.F.	165,00
12	A10025	87-116	10-04-87	.F.	1500,00

Hier ziet u een overzicht van de indexsleutels die voor het bestand Transact.dbf worden gebruikt:

Bijbehorende MDX-bestand:	TRANSACT.MDX	
Indexlabel:	KLANT_NR	Sleutel: KLANT_NR
Indexlabel:	BESTEL_NR	Sleutel: BESTEL_NR

Magazijn.dbf

Structuur van database: MAGAZIJN.DBF
Aantal records: 17
Laatst gewijzigd op : 17-10-88

Num	Veldnaam	Type	Breed	Dec	Index
1	BESTEL_NR	Teken	6		J
2	ONDRDL_NR	Teken	10		N
3	ARTIKEL	Teken	21		J
4	OMSCHRIJV	Teken	30		N
5	STUKPRIJS	Numeriek	8	2	N
6	AANTAL	Numeriek	3		N

** Totaal ** 81

Recordnr.	BESTEL_NR	ONDRDL_NR	ARTIKEL	OMSCHRIJV	STUKPRIJS	AANTAL
1	87-105	C-111-6010	SOFA,6 POTEN	LEDER,BRUIN,HOGE RUGLEUNING	1200,00	1
2	87-105	C-111-6015	SOFA,6 POTEN	FLUWEEL,GRIJS,FRANS	650,00	1
3	87-106	C-111-8050	SOFA,8 POTEN	FLUWEEL,BLAUW,FRANS	1200,00	1
4	87-107	C-222-1000	STOEL,BUREAU	LEDER,BRUIN,HOGE RUGLEUNING	1250,00	1
5	87-108	C-222-1000	STOEL,BUREAU	LEDER,BRUIN,HOGE RUGLEUNING	1250,00	1
6	87-109	C-400-2060	TAFEL,BIJZET	EIKENHOUT,2 POTEN,VIERKANT	250,00	1
7	87-109	C-500-6050	LAMP,STAAND	MESSING,6 POTEN,ENGELS	165,00	1
8	87-110	C-700-2020	DOSSIERKAST,2 LADEN	METAAL,BRUIN	75,00	1
9	87-110	C-700-4020	DOSSIERKAST,4 LADEN	METAAL,BRUIN	100,00	1
10	87-111	C-222-1001	STOEL,BUREAU	LEDER,BRUIN	1000,00	1
11	87-112	C-222-3020	STOEL,KANTINE	PLASTIC,GRIJS	350,00	2
12	87-113	C-300-2020	BOEKENKAST	TEAKHOUT,2 PLANKEN	125,00	1
13	87-114	C-500-6000	LAMP,STAAND	MESSING,6 POTEN,ART DECO	150,00	3
14	87-115	C-500-6050	LAMP,STAAND	MESSING,6 POTEN,ENGELS	165,00	1
15	87-116	C-600-5000	BUREAU,DIRECTIE,5 POTEN	EIKENHOUT,FINEER	1500,00	1
16	87-116	C-700-2030	LADENKAST,2 LADEN	METAAL,ZWART	75,00	1
17	87-116	C-222-1001	STOEL,BUREAU	LEDER,BRUIN	1000,00	1

Hier ziet u een overzicht van de indexsleutels die voor het bestand Magazijn.dbf worden gebruikt:

Bijbehorend MDX-bestand:	MAGAZIJN.MDX	
Indexlabel:	BESTEL_NR	Sleutel: BESTEL_NR
Indexlabel:	ARTIKEL	Sleutel: ARTIKEL

Menus.prg

Het volgende is een uitdraai van Menus.prg. Door het programma worden een horizontaal balkmenu en vier pop-up menu's gedefinieerd.

```
* Menus.prg
SET TALK OFF
CLEAR
Mwijzig = .F.
DO Def_menu
ON PAD View OF Hoofd ACTIVATE POPUP Visie_pop
ON PAD Goto OF Hoofd ACTIVATE POPUP Ganaar_pop
ON PAD Print OF Hoofd ACTIVATE POPUP Print_pop
ON SELECTION PAD Exit OF Hoofd ACTIVATE POPUP Stop_pop
ON SELECTION POPUP Stop_pop DO Stop_pro
ON SELECTION POPUP Visie_pop DO Visie_pro
ACTIVATE MENU Hoofd PAD View
CLEAR ALL
SET TALK ON
RETURN

PROCEDURE Def_menu
*—— Hoofdmenu
DEFINE MENU Hoofd
DEFINE PAD View OF Hoofd PROMPT "Toevoegen/Bewerken" AT 2,4
DEFINE PAD Goto OF Hoofd PROMPT "Ganaar/Zoek" AT 2,16
DEFINE PAD Print OF Hoofd PROMPT "Afdrukken" AT 2,30
DEFINE PAD Exit OF Hoofd PROMPT "Stop" AT 2,38
*—— Popup Visie_pop
DEFINE POPUP Visie_pop FROM 3,4 TO 8,19
DEFINE BAR 1 OF Visie_pop PROMPT "Nieuw record toevoegen"
DEFINE BAR 2 OF Visie_pop PROMPT "Bewerken"
DEFINE BAR 3 OF Visie_pop PROMPT "REPLICATE("-",16) SKIP"
DEFINE BAR 4 OF Visie_pop PROMPT "Wissen SKIP FOR Mwijzig"
*—— Popup Ganaar_pop
DEFINE POPUP Ganaar_pop FROM 3,16 TO 6,28
DEFINE BAR 1 OF Ganaar_pop PROMPT "Overslaan"
DEFINE BAR 2 OF Ganaar_pop PROMPT "Spring naar"
*—— Popup Print_pop
DEFINE POPUP Print_pop FROM 3,30 TO 7,42
DEFINE BAR 1 OF Print_pop PROMPT "Bestemming"
DEFINE BAR 2 OF Print_pop PROMPT "Opties"
DEFINE BAR 3 OF Print_pop PROMPT "Pagina doorvoeren"
*—— Popup Stop_pop
DEFINE POPUP Stop_pop FROM 3,38 TO 6,57
DEFINE BAR 1 OF Stop_pop PROMPT "Stop"
DEFINE BAR 2 OF Stop_pop PROMPT "Naar commandostip"
RETURN
```

```
PROCEDURE Stop_pro
DO CASE
  CASE BAR() = 1
          QUIT
          CANCEL
   CASE BAR() = 2
          DEACTIVATE MENU
ENDCASE
RETURN

PROCEDURE Visie_pro
DO CASE
  CASE BAR() = 1
          APPEND BLANK
          EDIT NEXT 1
  CASE BAR() = 2
          EDIT NEXT 1
           Mwijzig = .NOT. Mwijzig
  CASE BAR() = 4
          DELETE
ENDCASE
RETURN
* EOP: Menus.prg
```

Bestandsnaamextensies

In het eerste gedeelte van deze bijlage treft u alle bestandsnaamextensies met een korte omschrijving aan. In het tweede gedeelte wordt de relatie tussen ongecompileerde en gecompileerde bestanden vermeld. Tevens wordt aangegeven of deze bestanden compatibel zijn met dBASE III PLUS.

Bestandsnaamextensies in dBASE IV

In dBASE IV bestaat de extensie achter de bestandsnaam en de punt uit twee of drie letters. U kunt alle geldige DOS-bestandsnamen opgeven (raadpleeg hoofdstuk 1, "Basisbegrippen", voor een volledige beschrijving van bestandsnamen). In tabel D-1 worden de extensies vermeld en wordt de bestandsinhoud kort omschreven.

Tabel D-1 Bestandsnaamextensies in dBASE IV

Extensie	Bestandsinhoud
.$$$	Tijdelijk bestand; onvoltooide handeling
.app	Objectbestand van toepassingsontwerp; alleen in applicatiegenerator
.bak	Reservekopie van programma- (.prg) of procedure-bestand
.bar	Ontwerpobjectbestand van horizontale balk; alleen in applicatiegenerator
.bch	Ontwerpobjectbestand van batch-proces; alleen in applicatiegenerator
.bin	Binair bestand
.cat	Catalogusbestand
.cht	Bestand van CHART-MASTER®; wordt gebruikt voor dBASE/CHART-MASTER Bridge
.cod	Template-bronbestand
.cpt	Gecodeerd memobestand; wordt gebruikt bij wachtwoordbestand (.crp)
.crp	Wachtwoordbestand; wordt alleen met PROTECT gemaakt
.cvt	CONVERT-bestand; een bestand waarin de wijzigingen aan bestanden in een multi-user omgeving kunnen worden opgespoord
.db	Configuratiebestand; voor de standaardinstellingen bij het starten van dBASE IV
.db2	Oud dBASE II-bestand dat opnieuw is benoemd; wordt gebruikt voor het importeren en het exporteren

(wordt vervolgd)

Tabel D-1 Bestandsnaamextensies in dBASE IV *(vervolg)*

Extensie	Bestandsinhoud
.dbf	Database-bestand
.dbo	Objectbestand van commando- of procedurebestand
.dbt	Memobestand van database
.def	Definitiebestand van selector
.dif	DIF-bestand (Data Interchange Format) of VisiCalc-bestand; wordt gebruikt bij APPEND FROM/COPY
.doc	Documentatiebestand; alleen in applicatiegenerator
.fil	Ontwerpobjectbestand van bestandenlijst
.fmo	Gecompileerd indelingsbestand (.fmt)
.fmt	Gegenereerd indelingsbestand; uit .scr-bestand
.fr3	Oud dBASE III-rapportmodelbestand (.frm) dat opnieuw is benoemd
.frg	Gegenereerd rapportmodelbestand; uit .frm-bestand
.frm	Rapportmodelbestand
.fro	Gecompileerd rapportmodelbestand (.frg)
.fw2	Spreadsheet/database-bestand van Framework; wordt gebruikt voor het importeren en het exporteren
.gen	Template-bestand
.key	Bibliotheekbestand van toetsmacro's
.lb3	Oud dBASE III-etiketmodelbestand (.lbl) dat opnieuw is benoemd
.lbg	Gegenereerd etiketmodelbestand; uit .lbl-bestand
.lbl	Etiketmodelbestand
.lbo	Gecompileerd etiketmodelbestand (.lbl)
.log	Log-bestand van transactie
.mdx	Meervoudig indexbestand
.mem	Bestand met geheugenvariabelen
.ndx	Enkelvoudig indexbestand
.npi	Rapport-, formulier-, etiketbestanden; alleen in template-interpreter
.out	Bestand met uitvoer van commando's uit de template-taal
.pop	Ontwerpobjectbestand van pop-up menu; alleen in applicatiegenerator
.pr2	Bestand met printerstuurprogramma
.prf	Afdrukmodelbestand
.prg	Programma- of procedurebestand van dBASE

(wordt vervolgd)

Tabel D-1 Bestandsnaamextensies in dBASE IV *(vervolg)*

Extensie	Bestandsinhoud
.prs	Commando- of procedurebestand van dBASE/SQL
.prt	Printeruitvoerbestand
.qbe	QBE query-bestand
.qbo	Gecompileerd QBE query-bestand (.qbe)
.qry	Query-bestand
.rpd	RapidFile-bestand; wordt gebruikt voor het importeren en exporteren en voor APPEND FROM en COPY
.sc3	Oud dBASE III-schermbestand (.scr) dat opnieuw is benoemd

Structuur van een database-bestand (.dbf)

Bestandsaanhef en records van een database

Een database-bestand is samengesteld uit een bestandsaanhef, records met gegevens, wismarkeringen en een einde-bestandmarkering. De bestandsaanhef bevat gegevens over de bestandsstructuur en de records bevatten de eigenlijke gegevens. Van elk record is één byte gereserveerd voor de wismarkering.

Structuur van een database-bestandsaanhef

De bestandsaanhef, waarvan de bijzonderheden in tabel E-1 en E-2 worden vermeld, bevat gegevens die dBASE IV voor het bijhouden van het database-bestand nodig heeft.

Tabel E-1 Bestandsaanhef van een database-bestand

Byte	Inhoud	Betekenis
0	1 byte	Geldig dBASE IV-bestand. De bits 0-2 duiden het versienummer aan, de bits 3-5 zijn gereserveerd voor SQL en de bits 6 en 7 geven aan of er een memobestand aanwezig is.
1-3	3 bytes	Datum laatste mutatie. Notatie: DDMMJJ
4-7	32-bits getal	Aantal records in het database-bestand
8-9	16-bits getal	Aantal bytes in de bestandsaanhef
10-11	16-bits getal	Aantal bytes in het record
12-13	2 bytes	Gereserveerd
14	1 byte	Markering ter aanduiding van onvolledige transactie*
15	1 byte	Markering gecodeerd bestand**
16-27	12 bytes	Gereserveerd voor dBASE IV in een lokaal netwerk (LAN)
28	1 byte	Markering bijbehorend indexbestand (.mdx). 01H als er een bijbehorend indexbestand (.mdx) is, 00H indien dit niet aanwezig is.
29-31	3 bytes	Gereserveerd
32-n***	32 bytes elk	Veldbeschrijvingsreeks (de structuur van deze reeks ziet u in tabel E-2).
n + 1	1 byte	0DH als het veldbeëindigingsteken

* Deze markering wordt door de functie ISMARKED() gecontroleerd. Met BEGIN TRANSACTION wordt deze op 01 ingesteld en met END TRANSACTION en ROLLBACK weer op 00 ingesteld.

** Als deze markering op 01H is ingesteld, wordt de melding **Database is gecodeerd** weergegeven. Wanneer deze markering in 00H wordt gewijzigd, wordt de melding verwijderd maar wordt het bestand niet gedecodeerd.

*** n is de laatste byte van de veldbeschrijvingsreeks. De grootte van de reeks is afhankelijk van het aantal velden in het database-bestand

Tabel E-2 Bytes voor beschrijving van database-veld

Byte	Inhoud	Betekenis
0-10	11 bytes	Veldnaam in ASCII (met nullen gevuld)
11	1 byte	Veldtype in ASCII (C, D, F, L, M of N)
12-15	4 bytes	Gereserveerd
16	1 byte	Binaire notatie van veldlengte
17	1 byte	Binaire notatie van decimale veldtelling
18-19	2 bytes	Gereserveerd
20	1 byte	Naam van het werkgebied
21-31	11 bytes	Gereserveerd

Database-records

In het database-bestand komen de records na de bestandsaanhef. Gegevensrecords worden voorafgegaan door één byte. Dit wil zeggen: een spatie (20H) als het record niet is verwijderd en een asterisk (2AH) als het record is verwijderd. Velden worden zonder veldscheidingstekens of recordbeëindigingstekens in records samengebracht. Het einde van het bestand wordt door één byte gemarkeerd: met het ASCII-teken 26 (1AH) als einde-bestandmarkering.

ASCII-gegevens kunt u invoeren zoals dit in tabel E-3 wordt aangegeven.

Tabel E-3 Toegestane invoer voor elk gegevenstype

Gegevenstype	Toegestane invoer
C (teken)	Alle ASCII-tekens
D (datum)	Getallen en een teken waarmee dag, maand en jaar worden gescheiden (intern opgeslagen als 8 cijfers in de notatie JJJJMMDD)
F (binair numeriek met zwevende komma)	- , 0 1 2 3 4 5 6 7 8 9
L (logisch)	? Y y N n T t F f (? wanneer niet geïnitialiseerd)
M (memo)	Alle ASCII-tekens (intern opgeslagen als 10 cijfers die een bloknummer van .dbt voorstellen)
N (binair-gecodeerd decimaal numeriek getal)	- , 0 1 2 3 4 5 6 7 8 9

Memovelden en het .dbt-bestand

Een memobestand (.dbt) bestaat uit blokken die sequentieel zijn genummerd (0, 1, 2 enzovoort). Met SET BLOCKSIZE wordt de grootte van elk blok bepaald. Het eerste blok van het memobestand, blok 0, is de bestandsaanhef van het memobestand.

Elk memoveld van elk record in het .dbf-bestand bevat het nummer van het blok (in ASCII) waar het memoveld begint. Als zich in het memoveld geen gegevens bevinden, bevat het .dbf-bestand spaties (20H) en geen nummer.

Wanneer gegevens in een memoveld worden gewijzigd, kunnen de bloknummers ook worden gewijzigd. Tevens kan het nummer in het .dbf-bestand worden gewijzigd om de nieuwe lokatie aan te duiden.

In tegenstelling tot dBASE III PLUS kan dBASE IV de ruimte, die ontstaat wanneer u tekst uit een memoveld verwijderd, opnieuw gebruiken als u nieuwe tekst invoert. Door dBASE III PLUS werd nieuwe tekst altijd aan het einde van het .dbt-bestand toegevoegd. In dBASE III PLUS werd het .dbt-bestand steeds groter als nieuwe tekst werd toegevoegd, ook als tekst uit het bestand was verwijderd.

Printerstuurprogramma's van dBASE IV

De configuratie van dBASE IV kunt u zodanig instellen dat u met de meeste printers kunt afdrukken. U doet dit door de bestanden met de printerstuurprogramma's die zich op de installatiediskette bevinden, te installeren. Aan de hand van de vragen die het installatieprogramma op het scherm weergeeft, kunt u opgeven welke printer u gebruikt. Daarna wordt het bijbehorende bestand met het printerstuurprogramma geïnstalleerd. U kunt ook de standaard-printerstuurprogramma's in het bestand Config.db wijzigen of vanaf de commandostip _pdriver = <bestandsnaam van printerstuurprogramma> opgeven. Deze handelingen kunt u op elk gewenst moment na de installatie verrichten.

dBASE IV ondersteunt in Config.db maximaal vier verschillende printerstuurprogramma's met elk vijf lettertypen. Raadpleeg *Aan de slag met dBASE IV* voor informatie over het bestand Config.db en hoofdstuk 5, "Systeemgeheugenvariabelen", voor informatie over de systeemgeheugenvariabele _pdriver.

In de volgende tabel ziet u de bestanden met printerstuurprogramma's die zich op de installatiediskette bevinden. Tevens worden in deze tabel gegevens over de lettertypen die elk printerstuurprogramma ondersteunt, vermeld. Wanneer bij een printerstuurprogramma wordt vermeld dat *kadertekens* worden ondersteund, wil dat zeggen dat met dat printerstuurprogramma de ASCII-tekens met decimale waarden tussen 169 en 218 kunnen worden afgedrukt. Als het printerstuurprogramma geen kadertekens ondersteunt, worden kaders met streepjes (ASCII 45) en pipeline-tekens (ASCII 124) afgedrukt.

Tabel F-1 Printerstuurprogramma's van dBASE IV

Fabrikant	Model	Bestandsnaam printerstuurprogramma
AMT	AMT 250	AMT.pr2
Apple	LaserWriter[1]	POSTSCRI.pr3 POSTSCRI.dld[2]
Anadex	6000 9625B (Silent Scribe)	ANA6000.pr2 ANA9625B.pr2
ASCII[3]		ASCII.pr2
Brother	HR-15, HR-25	HR15.pr2
Canon USA, Inc.	A2 in grafische modus 150 dpi	CANA2_15.pr2
C.Itoh Digital Products, Inc.	1550A, B, BPI 1550S, SC 715F 815 (P351 Mode) 8510A, B, BPI 8510S, SC 3500 model 20 ProWriter Jr.	CI1550A.pr2 CI1550S.pr2 CI715_2.pr2 CI815.pr2 CI8510A.pr2 CI8510S.pr2 CI3520.pr2 FX80_1.pr2
Datasouth	180 220	DS180.pr2 DS220.prR
DataProducts	P-80, P-132 SPG-8050, SPG-8070	DP_P80.pr2 DP_8050.pr2
Diablo	(Xerox) 630 API	DIAB630A.pr2
Epson America, Inc.	FX-85, FX-185 FX-86E GQ-3500 JX-80 JX-80 met 8177 LX-80, LX-90 MX-80, MX-100 MX-80/MX-100 met Graftrax PLUS RX-80, FX-80, FX-80 PLUS RX-80/FX-80 met 8177 RX-100, FX-100, FX-100 PLUS RX-100/FX-100 met 8177 LQ-800 LQ-1500, SQ-1500 DX-10 DX-20, DX-35	FX85_1.pr2 FX86E_1.pr2 GQ35_150.pr2 JX80_2.pr2 EPSN8177.pr2 LX80.pr2 Generic.pr2 MX80G.pr2 FX80_1.pr2 EPSN8177.pr2 FX80_1.pr2 EPSN8177.pr2 LQ800_2.pr2 LQ1500_1.pr2 DX10.pr2 DIAB630A.pr2
Facit	4512	FAC4512.pr2
Fujitsu	DotMax 9f DotMax 9i DotMax 24i DotMAx 24c	FX80_1.pr2 IBMGP.pr2 FUJ24I.pr2 FUJ24C.pr2
Generic-printerstuurprogramma[4]	Iedere niet vermelde printer	GENERIC.pr2

1 Of een volledig compatibele printer die Adobe PostScript gebruikt.

2 Het .dld-bestand is een programma dat vanaf het systeem wordt geladen.

3 Met het ASCII-printerstuurprogramma kunt u ASCII-bestanden maken waarin geen printer escape-codes voorkomen.

4 Het Generic-printerstuurprogramma levert elementaire opmaakopties voor een breed assortiment printermodellen.

Kader-tekens	Vet	Cursief	Onder-strepen	Elite	Gecom-primeerd	Correspondentie-kwaliteit
Nee	Dubbel	Ja	Streepjes	Nee	Nee	Ja
Nee	Ja	Ja	Ja	Ja	Ja	n.v.t.
Nee	Dubbel	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Streepjes	Streepjes	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	Nee
Nee	Ja	Ja	Ja	Ja	Ja	Nee
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	Nee
Nee	Ja	Ja	Ja	Ja	Ja	Nee
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Onderstrepen	Streepjes	Ja	Ja	Nee
Nee	Dubbel	Streepjes	Streepjes	Nee	Ja	Ja
Ja	Dubbel	Streepjes	Streepjes	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Nee	Nee	n.v.t.
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Streepjes	Streepjes	Nee	Nee	Nee
Nee	Ja	Nee	Ja	Nee	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Nee	Nee	Nee
Nee	Ja	Onderstrepen	Ja	Nee	Nee	n.v.t.
Nee	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Dubbel	Streepjes	Streepjes	Nee	Nee	Nee

Tabel F-1 Printerstuurprogramma's van dBASE IV *(vervolg)*

Fabrikant	Model	Bestandsnaam printerstuurprogramma
Hewlett-Packard, Inc.	LaserJet	
	LJ 6.5 lpi[5]	HPLAS100.pr2
	LJ 6 lpi[6]	HPLAS60.pr2
	LJ land 6.5 lpi[7]	HPLASL.pr2
	LJ land 6 lpi[8]	HPLASL45.pr2
	LJII 6.5 lpi, LJIID1 6.5 lpi (enkelzijdig)	HPLAS2I.pr2
	LJII 6 lpi, LJIID1 6 lpi (enkelzijdig)[6]	HPLAS260.pr2
	LJII land 6 lpi[8]	HPLS2L45.pr2
	LJIID2 6.5 lpi (tweezijdig)	HPLASID.pr2
	LJIID2 6 lpi (tweezijdig)[6]	HPLASD60.pr2
	LJIID1 6.5 lpi (enkelzijdig)	HPLAS2I.pr2
	DeskJet (HP2276)	HPDKS150.pr2
	PaintJet	HPPAINT.pr2
	QuietJet	HPQJET1.pr2
	ThinkJet	HPJET.pr2
International Business Machines Corporation	Color JetPrinter	IBMCJET2.pr2
	Graphics Printer	IBMGP.pr2
	QuietWriter (model 1)	IBMQUIET.pr2
	QuietWriter (model 2)	IBMQ2_1.pr2
	QuietWriter (model 3)	IBMQ3_1.pr2
	ProPrinter	IBMPRO_1.pr2
	WheelPrinter	IBMWHEEL.pr2
Nippon Electric Corporation	Pinwriter P2, P3	IBMGP.pr2
	Pinwriter P5	NECP5.pr2
	Spinwriter 3550	NEC3550.pr2
	Spinwriter 7710	NEC7710.pr2
	Spinwriter 8850	NEC8850.pr2
Oki America, Inc.	Microline 80, 82, 83A	GENERIC.pr2
	Microline 82A/83A met Okigraph	OKI82AGR.pr2
	Microline 84	OKI84.pr2
	Microline 182, 183	OKI182.pr2
	182/183 met Plug `n Play	OKI182PN.pr2
	Microline 92, 93	OKI92_2.pr2
	Microline 192, 193	OKI192_2.pr2
	92/93/192/193 met Plug `n Play	OKI92PNP.pr2
	Okimate-20	IBMGP.pr2
	Pacemark 2350, 2410	OKI2410.pr2
Qume	Sprint 11 Plus, LetterPro 20	QUME11.pr2
Ricoh	6000 Portrait Orient	RIC6K10.pr2
	6000 Landscape Orient	RIC6K10L.pr2
Star Micronics	Gemini-10x, Gemini-15x	GEM10.pr2
Tandy Corporation	CGP-220	CGP220.pr2
Texas Instruments, Inc.	855, 865	TI855.pr2
Toshiba America, Inc.	P321, P341, P351	P351.pr2
	P1340, P1351	P1340.pr2
	PageLaser/2	TOSLAS15.pr2
Xerox Corporation	(Diablo) 630 API	DIAB630A.pr2

5 lpi = lines per inch
6 Paginalengte ingesteld op 60
7 Paginalengte ingesteld op 51
8 Paginalengte ingesteld op 45

Kader-tekens	Vet	Cursief	Onder-strepen	Elite	Gecom-primeerd	Correspondentie-kwaliteit
Nee	Dubbel	Onderstrepen	Ja	Ja	Nee	n.v.t.
Nee	Dubbel	Onderstrepen	Ja	Ja	Nee	n.v.t.
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Ja	Ja	Ja	Ja	Ja	Ja	n.v.t.
Nee	Ja	Ja	Ja	Ja	Ja	n.v.t.
Nee	Ja	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Ja	Onderstrepen	Ja	Ja	Ja	Nee
Nee	Ja	Onderstrepen	Ja	Nee	Ja	Nee
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Ja	Dubbel	Onderstrepen	Ja	Nee	Nee	Nee
Ja	Dubbel	Onderstrepen	Ja	Nee	Nee	Nee
Ja	Ja	Onderstrepen	Ja	Ja	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Ja	Ja	Ja
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	n.v.t.
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Ja	Ja	Ja	Streepjes	Ja	Ja	Ja
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Ja	Onderstrepen	Ja	Nee	Nee	n.v.t.
Nee	Ja	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Dubbel	Onderstrepen	Ja	Nee	Nee	Nee
Nee	Dubbel	Streepjes	Streepjes	Nee	Ja	Nee
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	Ja
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	Ja
Ja	Ja	Ja	Ja	Ja	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Ja	Ja	Onderstrepen	Ja	Nee	Ja	Ja
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	Ja
Nee	Ja	Onderstrepen	Ja	Ja	Ja	n.v.t.
Nee	Ja	Onderstrepen	Ja	Nee	Ja	n.v.t.
Ja	Ja	Onderstrepen	Ja	Nee	Ja	n.v.t.
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Nee	Nee	Nee	Ja	Nee	Nee
Nee	Ja	Ja	Streepjes	Ja	Ja	Ja
Nee	Ja	Ja	Ja	Ja	Ja	Ja
Nee	Dubbel	Onderstrepen	Ja	Ja	Ja	Ja
Ja	Ja	Ja	Ja	Nee	Ja	n.v.t.
Nee	Ja	Onderstrepen	Ja	Nee	Nee	Nee

ASCII-tabel

Binair	Hex	Decimaal	Teken	Code	Symbool	Omschrijving
00000000	00	0		^@	NUL	Nulteken
00000001	01	1	☺	^A	SOH	Beginteken van de aanhef
00000010	02	2	●	^B	STX	Beginteken van de tekst
00000011	03	3	♥	^C	ETX	Einde-tekstteken
00000100	04	4	♦	^D	EOT	Einde-transmissieteken
00000101	05	5	♣	^E	ENQ	Navraagteken
00000110	06	6	♠	^F	ACK	Bevestigingsteken
00000111	07.	7	•	^G	BEL	Bel
00001000	08	8	◘	^H	BS	Spatie-terugteken
00001001	09	9	○	^I	SH	Horizontaal tabelleerteken
00001010	0A	10	◙	^J	LF	Regeldoorvoerteken
00001011	0B	11	♂	^K	VT	Verticaal tabelleerteken
00001100	0C	12	♀	^L	FF	Paginadoorvoerteken
00001101	0D	13	♪	^M	CR	Wagenterugloopteken
00001110	0E	14	♫	^N	SO	SHIFT-OUT
00001111	0F	15	☼	^O	SI	SHIFT-IN
00010000	10	16	►	^P	DLE	Wisselteken voor bewerkingsuitbreiding
00010001	11	17	◄	^Q	DC1	Besturingsteken voor randapparatuur 1
00010010	12	18	↕	^R	DC2	Besturingsteken voor randapparatuur 2
00010011	13	19	‼	^S	DC3	Besturingsteken voor randapparatuur 3
00010100	14	20	¶	^T	DC4	Besturingsteken voor randapparatuur 4
00010101	15	21	§	^U	NAK	Negatief bevestigingsteken
00010110	16	22	■	^V	SYN	Synchronisatieteken
00010111	17	23	↨	^W	ETB	Einde-transmissieblokteken
00011000	18	24	↑	^X	CAN	Annuleringsteken
00011001	19	25	↓	^Y	EM	Einde-mediumteken

Binair	Hex	Decimaal	Teken	Code	Symbool	Omschrijving
00011010	1A	26	→	^Z	SUB	Vervangingsteken
00011011	1B	27	←	^[	ESC	Escape
00011100	1C	28	⌙	^|	FS	Bestandsscheidingsteken
00011101	1D	29	↔	^]	GS	Groepenscheidingsteken
00011110	1E	30	▲	^^	RS	Recordscheidingsteken
00011111	1F	31	▼	^-	US	Eenhedenscheidingsteken
00100000	20	32				
00100001	21	33	!			
00100010	22	34	“			
00100011	23	35	#			
00100100	24	36	$			
00100101	25	37	%			
00100110	26	38	&			
00100111	27	39	‘			

Binair	Hex	Decimaal	Teken	Binair	Hex	Decimaal	Teken
00101000	28	40	(	00110111	37	55	7
00101001	29	41	)	00111000	38	56	8
00101010	2A	42	*	00111001	39	57	9
00101011	2B	43	+	00111010	3A	58	:
00101100	2C	44	‘	00111011	3B	59	;
00101101	2D	45	-	00111100	3C	60	<
00101110	2E	46	.	00111101	3D	61	=
00101111	2F	47	/	00111110	3E	62	>
00110000	30	48	0	00111111	3F	63	?
00110001	31	49	1	01000000	40	64	@
00110010	32	50	2	01000001	41	65	A
00110011	33	51	3	01000010	42	66	B
00110100	34	52	4	01000011	43	67	C
00110101	35	53	5	01000100	44	68	D
00110110	36	54	6	01000101	45	69	E

Binair	Hex	Decimaal	Teken
01000110	46	70	F
01000111	47	71	G
01001000	48	72	H
01001001	49	73	I
01001010	4A	74	J
01001011	4B	75	K
01001100	4C	76	L
01001101	4D	77	M
01001110	4E	78	N
01001111	4F	79	O
01010000	50	80	P
01010001	51	81	Q
01010010	52	82	R
01010011	53	83	S
01010100	54	84	T
01010101	55	85	U
01010110	56	86	V
01010111	57	87	W
01011000	58	88	X
01011001	59	89	Y
01011010	5A	90	Z
01011011	5B	91	[
01011100	5C	92	\
01011101	5D	93	]
01011110	5E	94	^
01011111	5F	95	_
01100000	60	96	\`
01100001	61	97	a
01100010	62	98	b
01100011	63	99	c
01100100	64	100	d
01100101	65	101	e

Binair	Hex	Decimaal	Teken
01100110	66	102	f
01100111	67	103	g
01101000	68	104	h
01101001	69	105	i
01101010	6A	106	j
01101011	6B	107	k
01101100	6C	108	l
01101101	6D	109	m
01101110	6E	110	n
01101111	6F	111	o
01110000	70	112	p
01110001	71	113	q
01110010	72	114	r
01110011	73	115	s
01110100	74	116	t
01110101	75	117	u
01110110	76	118	v
01110111	77	119	w
01111000	78	120	x
01111001	79	121	y
01111010	7A	122	z
01111011	7B	123	{
01111100	7C	124	|
01111101	7D	125	}
01111110	7E	126	o
01111111	7F	127	⌂
10000000	80	128	Ç
10000001	81	129	ü
10000010	82	130	é
10000011	83	131	â
10000100	84	132	ä
10000101	85	133	à

Binair	Hex	Decimaal	Teken
10000110	86	134	å
10000111	87	135	ç
10001000	88	136	ê
10001001	89	137	ë
10001010	8A	138	è
10001011	8B	139	ï
10001100	8C	140	î
10001101	8D	141	ì
10001110	8E	142	Ä
10001111	8F	143	Å
10010000	90	144	É
10010001	91	145	æ
10010010	92	146	Æ
10010011	93	147	ô
10010100	94	148	ö
10010101	95	149	ò
10010110	96	150	û
10010111	97	151	ù
10011000	98	152	ÿ
10011001	99	153	Ö
10011010	9A	154	Ü
10011011	9B	155	¢
10011100	9C	156	£
10011101	9D	157	¥
10011110	9E	158	₧
10011111	9F	159	ƒ
10100000	A0	160	á
10100001	A1	161	í
10100010	A2	162	ó
10100011	A3	163	ú
10100100	A4	164	ñ
10100101	A5	165	Ñ

Binair	Hex	Decimaal	Teken
10100110	A6	166	ª
10100111	A7	167	º
10101000	A8	168	¿
10101001	A9	169	⌐
10101010	AA	170	¬
10101011	AB	171	½
10101100	AC	172	¼
10101101	AD	173	¡
10101110	AE	174	«
10101111	AF	175	»
10110000	B0	176	░
10110001	B1	177	▒
10110010	B2	178	▓
10110011	B3	179	│
10110100	B4	180	┤
10110101	B5	181	╡
10110110	B6	182	╢
10110111	B7	183	╖
10111000	B8	184	╕
10111001	B9	185	╣
10111010	BA	186	║
10111011	BB	187	╗
10111100	BC	188	╝
10111101	BD	189	╜
10111110	BE	190	╛
10111111	BF	191	┐
11000000	C0	192	└
11000001	C1	193	┴
11000010	C2	194	┬
11000011	C3	195	├
11000100	C4	196	─
11000101	C5	197	┼

Teken	Binair	Hex	Decimaal
11000110	C6	198	╞
11000111	C7	199	╟
11001000	C8	200	╚
11001001	C9	201	╔
11001010	CA	202	╩
11001011	CB	203	╦
11001100	CC	204	╠
11001101	CD	205	═
11001110	CE	206	╬
11001111	CF	207	╧
11010000	D0	208	╨
11010001	D1	209	
11010010	D2	210	╥
11010011	D3	211	╙
11010100	D4	212	╘
11010101	D5	213	╒
11010110	D6	214	╓
11010111	D7	215	╫
11011000	D8	216	╪
11011001	D9	217	┘
11011010	DA	218	┌
11011011	DB	219	╹
11011100	DC	220	▄
11011101	DD	221	▌
11011110	DE	222	▐
11011111	DF	223	▀
11100000	E0	224	α
11100001	E1	225	β
11100010	E2	226	Γ
11100011	E3	227	π
11100100	E4	228	Σ
11100101	E5	229	σ
11100110	E6	230	μ
11100111	E7	231	τ
11101000	E8	232	ϕ
11101001	E9	233	θ
11101010	EA	234	Ω
11101011	EB	235	δ
11101100	EC	236	∞
11101101	ED	237	ø
11101110	EE	238	ε
11101111	EF	239	π
11110000	F0	240	≡
11110001	F1	241	±
11110010	F2	242	≥
11110011	F3	243	≤
11110100	F4	244	⌠
11110101	F5	245	⌡
11110110	F6	246	÷
11110111	F7	247	≈
11111000	F8	248	°
11111001	F9	249	•
11111010	FA	250	.
11111011	FB	251	√
11111100	FC	252	η
11111101	FD	253	2
11111110	FE	254	■
11111111	FF	255	

dBASE-commando's en -functies die in de SQL-modus zijn toegestaan

De SQL-taal bestaat uit een kleine set commando's en functies die uitsluitend dienen voor het vastleggen en het toegankelijk maken van gegevens. dBASE IV biedt een aanvulling op de SQL-commando's in de vorm van een set dBASE-commando's en -functies die u in SQL kunt gebruiken. In de interactieve modus kunt u bijvoorbeeld dBASE IV-commando's gebruiken om van gegevens die u met een SQL-query hebt geselecteerd, rapporten te maken en af te drukken. Ook kunt u met BROWSE door de resultaten van een SELECT-query bladeren.

In het volgende gedeelte vindt u een opsomming van de commando's en functies die u in de interactieve en de ingebedde SQL-modus kunt gebruiken.

dBASE-commando's die in de SQL-modus zijn toegestaan

?/??
???
@
@...CLEAR
@...FILL
@...TO
ACCEPT
ACTIVATE MENU
ACTIVATE POPUP
ACTIVATE SCREEN
ACTIVATE WINDOW
APPEND
ASSIST
BEGIN/END TRANSACTION
BROWSE
CALL
CANCEL
CHANGE
CLEAR
CLEAR GETS
CLEAR MEMORY
CLEAR MENUS
CLEAR POPUPS
CLEAR TYPEAHEAD
CLEAR WINDOWS
CLOSE
CLOSE ALTERNATE
CLOSE FORMAT
CLOSE PROCEDURE
COMPILE
COPY FILE
CREATE
CREATE FROM
CREATE/MODIFY APPLICATION
CREATE/MODIFY LABEL
CREATE/MODIFY QUERY/VIEW
CREATE/MODIFY REPORT
CREATE/MODIFY SCREEN
CREATE/MODIFY STRUCTURE
CREATE VIEW FROM ENVIRONMENT
DEACTIVATE MENU
DEACTIVATE POPUP
DEACTIVATE WINDOW
DEBUG
DECLARE
DEFINE BAR
DEFINE BOX
DEFINE MENU
DEFINE PAD
DEFINE POPUP
DEFINE WINDOW
DELETE FILE
DIR
DO
DO CASE
DO WHILE
EDIT

EJECT
EJECT PAGE
ENDPRINTJOB
END TRANSACTION
ERASE
EXIT
FUNCTION
HELP
IF
INPUT
KEYBOARD
LABEL FORM
LIST/DISPLAY FILES
LIST/DISPLAY HISTORY
LIST/DISPLAY MEMORY
LIST/DISPLAY STATUS
LIST/DISPLAY STRUCTURE
LIST/DISPLAY USERS
LOAD
LOGOUT
MODIFY COMMAND/FILE
MOVE WINDOW
NOTE/*/&&
ON ERROR/ESCAPE/KEY
ON PAD
ON PAGE
ON READERROR
ON SELECTION PAD
ON SELECTION POPUP
PACK
PARAMETERS
PLAY MACRO
PRINTJOB
PRIVATE
PROCEDURE
PROTECT
PUBLIC
QUIT
READ
RECALL
RELEASE
RELEASE MENUS
RELEASE MODULE
RELEASE POPUPS
RELEASE WINDOWS
RENAME
REPORT FORM
RESTORE
RESTORE MACROS
RESTORE SCREEN
RESTORE WINDOW
RESUME
RETRY
RETURN
SET PRECISION
RUN/!
SAVE
SAVE MACROS
SAVE SCREEN
SAVE WINDOW
SELECT
SET
SET ALTERNATE
SET AUTOSAVE
SET BELL
SET BORDER
SET CATALOG
SET CENTURY
SET CLOCK
SET COLOR
SET CONFIRM
SET CONSOLE
SET CURRENCY
SET CURSOR
SET DATE
SET DBTRAP
SET DEBUG
SET DECIMALS
SET DEFAULT
SET DELETED
SET DELIMITERS
SET DESIGN
SET DEVELOPMENT
SET DEVICE
SET DIRECTORY
SET DISPLAY
SET ECHO
SET ENCRYPTION
SET ESCAPE
SET EXACT
SET EXCLUSIVE
SET FORMAT
SET FULLPATH
SET FUNCTION
SET HEADING
SET HELP
SET HISTORY
SET HOURS
SET INSTRUCT
SET INTENSITY
SET LOCK
SET MARGIN
SET MARK
SET MESSAGE
SET NEAR
SET ODOMETER
SET PATH
SET PAUSE
SET POINT
SET TRAP

SET PRINTER
SET PROCEDURE
SET REFRESH
SET REPROCESS
SET SAFETY
SET SCOREBOARD
SET SEPARATOR
SET SPACE
SET SQL
SET STATUS
SET STEP
SET TALK
SET TITLE
SET TYPEAHEAD
SET UNIQUE
SET VIEW TO
SET WINDOW OF MEMO TO
SHOW MENU
SHOW POPUP
SORT TO
STORE
SUSPEND
TEXT
TYPE
USE
WAIT

dBASE-functies die in de SQL-modus zijn toegestaan

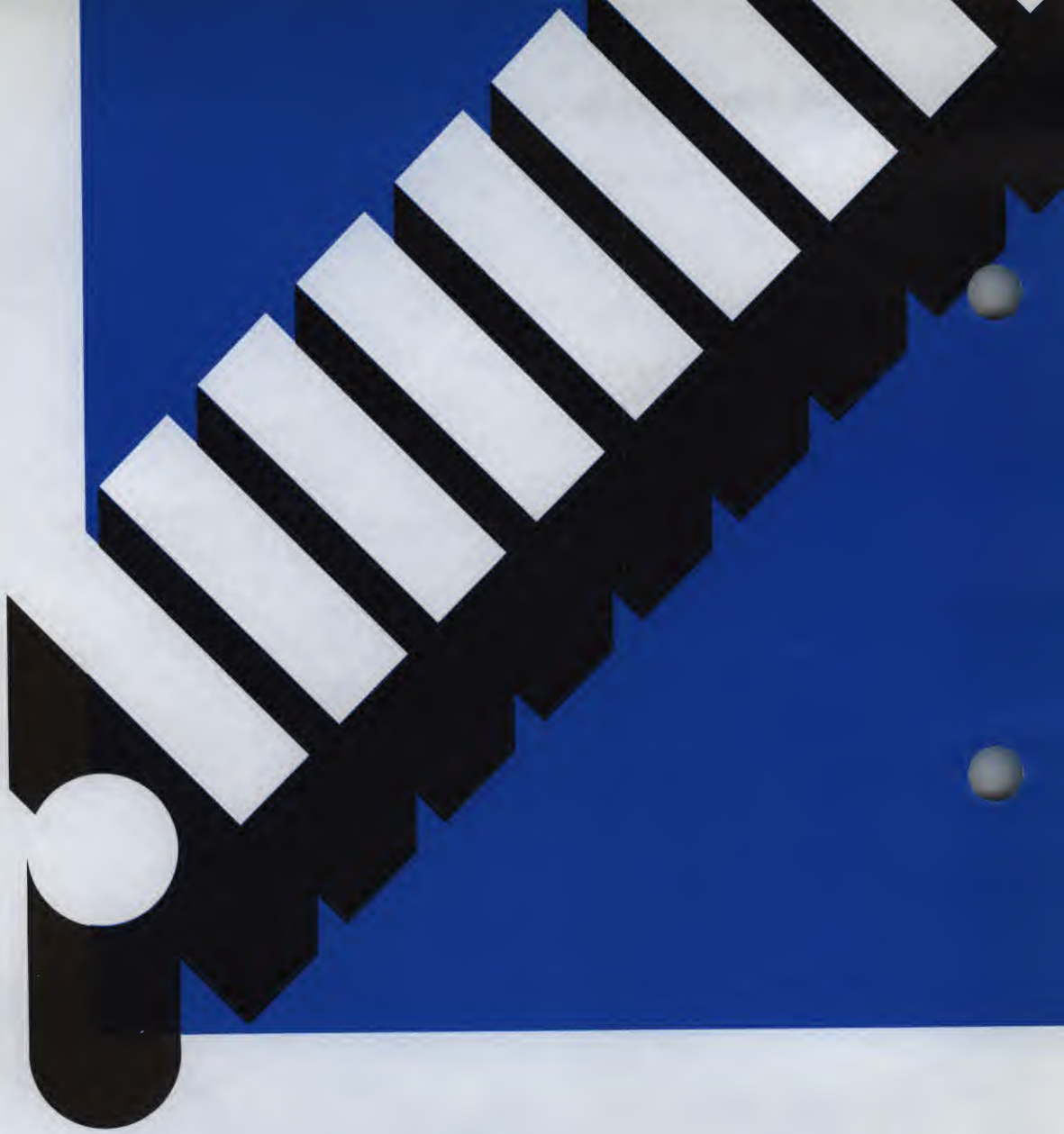
OPMERKING
Bepaalde dBASE-functies zijn weliswaar in de SQL-modus toegestaan, maar niet in een SQL-commando (bijvoorbeeld in een SELECT-commando of een clausule WHERE. Zulke functies zijn met een asterisk gemarkeerd.

&()*
ABS()
ACOS()
ALIAS()*
ASC()
ASIN()
AT()
ATAN()
ATN2()
BAR()*
CALL()*
CDOW()
CEILING()
CERROR()*
CHR()
CMONTH()
COL()*
COMPLETED()*
COS()
CTOD()
DATE()
DAY()
DIFFERENCE()
DISKSPACE()*
DMY()
DOW()
DTOC()
DTOR()
DTOS()
ERROR()*
EXP()
FILE()*
FIXED()
FKLABEL()*
FKMAX()*
FLOAT()
FLOOR()
GETENV()*
IIF()*
INKEY()*
INT()
ISALPHA()*
ISCOLOR()*
ISLOWER()*
ISUPPER()*
KEY()*
LASTKEY()*
LEFT()
LEN()
LIKE()*
LINENO()*
LOG()
LOG10()
LOWER()
LTRIM()
MAX()*
MDY()
MEMORY()*

MENU()*
MESSAGE()*
MIN()*
MOD()
MONT()
NETWORK()*
OS()*
PAD()*
PCOL()*
PI()
POPUP()*
PRINTSTATUS()*
PROGRAM()*
PROMPT()*
PROW()*
RAND()
READKEY()*
REPLICATE()
RIGHT()
ROLLBACK()*
ROUND()
ROW()*
RTOD()
RTRIM()
SELECT()*
SET()*
SIGN()
SIN()
SOUNDEX()
SPACE()
SQRT()
STR()
STUFF()
SUBSTR()
TAN()
TIME()
TRANSFORM()
TRIM()
TYPE()*
UPPER()
USER()
VAL()
VARREAD()*
VERSION()*
WINDOW()*
YEAR()

Index

X: Index



i 1 2 3 4 5 6 7 A B

C D E F G H In

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z

ASHTON-TATE®

10160NL0281